

Webbdatabas

ASP.NET + C#

Implementering –
skivdatabas administration

Av

Tommy Friberg

Innehållsförteckning

Dokumentation för webb-baserad skivdatabas	3
Introduktion	4
Installation-databas.....	5
Databasstruktur.....	6
Normaliseringsformer	7
Optimering	8
Applikationslogik.....	9
ASP.NET-specifika funktionaliteter	13
Inmatningsfält.....	16
Inledning.....	16
Knappar-funktionalitet	19
Inledning.....	19
Webbläsare och upplösningar	23
I användning.....	24
Framtida eventuella uppdateringar.....	25
KÄLLKOD.....	26
FRONTEND.....	28
Frontend - Aspx i par med aspx.cs - filer	29
Frontend - Usercontrols; GridViews och paneler i par med tillhörande cs-filer.....	89
Frontend - Themes	162
Frontend - JavaScript	164
Frontend - CSS-filer	167
MELLANSKIKTET.....	198
Mellanskiktet - Komponent: DB_Callcode.....	199
Mellanskiktet - Komponent: DB_Callcode_Artist.....	238
Mellanskiktet - Komponent: DB_Callcode_Inkop	269
Mellanskiktet - Komponent: DB_proc	306
Mellanskiktet - Komponent: DB_proc_Artist.....	316
Mellanskiktet - Komponent: DB_proc_Inkop.....	323
Mellanskiktet - Komponent: Common_Tasks	332
Mellanskiktet - Komponent: Common_Tasks_Artist	337
Mellanskiktet - Komponent: Common_Tasks_Inkop	341
Mellanskiktet - Komponent: CtrlLib.....	345
BACKEND.....	347
Backend - MySQL-databas	348
Bilagor.....	349
Bilaga 1 - Testschema	350
Bilaga 2 – Format på olika skivor/media som kan matas in i databasen	355

Dokumentation för webb-baserad skivdatabas



Bild1: Databasen som den ser ut när den körs under browsern Google Chrome 3.

Introduktion

Välkommen till en av de saker som jag länge haft i sinnet att förverkliga, nämligen en webbdatabas som kan användas för exempelvis som i det här fallet skivsamlingar. Den kan med vissa mindre modifieringar även användas för till exempel filmsamlingar eller andra snarlika behov. Jag har i många år använt Microsofts Access-databaser lokalt för detta ändamål och det har känts väldigt begränsat. Just att kunna nå ut med sin databas till fler människor eller i alla fall närmsta vänskretsen har ju varit en av grundtankarna med just ett gränssnitt för webben. Delad glädje är ju som det heter dubbel glädje och eftersom kultur är något mycket centralt i mitt liv så är det även något som jag vill dela med mig och nu kan jag göra det mer konkret än bara beskriva de artister, skivor och så vidare som jag uppskattar.

Just denna specifika ASP.NET variant av min webbdatabas har tagit den till nya höjder vilket har varit mycket jobb men samtidigt mycket lärorikt och spännande. Jag har sedan tidigare en ASP-databas och i och med att jag hade den så trodde jag i min enfald att det skulle vara relativt simpelt och inte ta så lång tid utan bara lägga till de nya ASP.NET-funktionerna och anpassa lite här och där. Invertera allt i den meningen så kommer vi nog en aning närmare sanningen. Det tog rejält med tid, mycket längre än jag trodde även om det så här efteråt kan säga att med så mycket arbete som det krävdes så gick det ändå snabbt. Jag läste någonstans innan jag började arbetet om att det skulle vara tuffare än man trodde att gå från ASP till ASP.NET och den som hade skrivit det visste verkligen vad den talade om. Men eftersom jag gillar utmaningen så antog jag den och som jag nämnda nyss, det har ändå retrospektivt sett inte tagit så lång tid med tanke på omfattningen. Själva utvecklingsarbetet tog exempelvis cirka 3,5 månader i anspråk vilket får ses som relativt bra. Därtill kommer ju mer tid för att lägga upp sidan på mitt webbhotell, som inte gick så smidigt samt att hotellet ofta lider av driftstörningar, så jag funderar faktiskt i framtiden i mån av pengar och så vidare att byta webbhotell, men får behålla skrället så länge. Även den här dokumentationen och den för client-delen tog en del tid att iordningsställa.

Några ord även om språket jag använt. Språket är det som brukar kallas för huvudspråket för ASP.NET, nämligen C#. Det är ett språk som nästan är gränslöst i sin omfattning och möjligheter och jag har nu skrapat något på dess yta i och med den här applikationen som ett tecken på hur långt jag har nått i skrivande stund. En slags skräckblandad förtjusning kan man säga att man blir frustrerad över att den är så vidsträckt så man troligtvis aldrig kommer lära sig alla dess aspekter men samtidigt fascinerande över de aspekter man hittills har upptäckt och de man kommer att upptäcka.

Installation-databas

Det som beskrivs här i den här dokumentationen är hur man sköter gränssnittet och dess kommunikation med databasen och inte själva databasen själv. Jag har för den här applikationen lämnat mina Accessdatabaser hädan och valt att istället jobba med MySQL som jag finner som ett gott val då den är tillämpningsbar på en rad plattformar och innehar tillräckligt med kapacitet och prestanda för den här typen av databas. Sedan att själva databasen är frikopplad från funktionalitet och eventuella affärsregler i mellanskiktet gör dess fysiska storlek mindre och därmed mer smidig.

Gällande installationen så är det som sagt inte egentligen den här dokumentationens uppgift att redogöra för den men för att ändå nämna något om denna process helt kort så installeras MySQL-databasprogrammet på den dator som skall agera som databas-server och som webbservern sedan skall kopplas upp mot. I mitt fall behövde jag också installera en drivrutin för det operativsystem jag har använt, Microsoft Windows XP Pro, för att det hela skulle fungera.

Därefter får man med hjälp av och via ett administrativt verktyg gå in och skapa själva databasen. Två grafiska sådana som jag använde och som kan rekommenderas är phpMyAdmin och SQLyog Enterprise. De kan även agera som ett bra komplement till denna administrativa applikation jag har konstruerat här. Det går också bra att arbeta direkt mot MySQL utan en grafisk front-end, med hjälp av MySQL:s egna DOS-baserade verktyg, om man föredrar att jobba direkt från en prompt.

När det är gjort så kan man antingen skapa de tre tabeller som denna skivdatabas består av (artist, cd och inkop) om det är första installationen. Om det inte är det så går det även att återskapa tabellen om man gjort en databasbackup innan till exempelvis en textfil, en så kallad logisk databasbackup. Då räcker det också med att bara manuellt först skapa själva databasen. Efter dessa åtgärder kan man återgå till den här applikationen och administrera databasen. Tänk bara på att det går inte på grund av tabellernas relationer att mata in en skiva om inte motsvarande artist först har lagts in i Artisttabellen.

För mer ingående detaljer gällande MySQL och eventuella grafiska administrationsverktyg, se deras respektive manualer eller hemsidor och dylikt på Internet.

Databasstruktur

Strukturen för den här databasen är tre tabeller en för skivor (som kallades cd men har sedermera expanderats till att även inkludera vinylskivor), en för förekommande artister samt en tredje för de inköp av skivor som skall tänkas göras. Dessa har mellan sig två stycken ett-till-många-förhållanden där en artist kan förekomma på många redan inköpta skivor samt en artist kan förekomma på många skivor som skall införskaffas genom inköp. De är rent konkret sammankopplade via en primärnyckel i Artisttabellen och en tillhörande främmande nyckel i Skiv- respektive Inkopstabellen. Detta medför rent praktiskt att man kan inte mata in en skiva, varken i Skiv- eller Inkopstabellen, av en artist som inte finns representerad i Artisttabellen. Samma sak om man tar bort en artist i Artisttabellen så försvinner de skivor som förekommer med honom både i Skiv- och Inkopstabellerna.

Normaliseringsformer

Beträffande normaliseringsformer så är min tolkning att databasen ligger någonstans mellan normaliseringsform två och tre. Denna osäkerhet beror på hur man tolkar de två fälten Ursland och Ursstad i Artisttabellen. För normaliseringsform två så ser jag inget fullständigt funktionellt beroende mellan dessa två. Ett land innehåller ju många olika städer men å andra sidan så ligger ju alltid en viss stad i ett visst land. Sedan skall ju även detta beroende relatera till primärnyckeln vilket jag inte alls kan se hur det gör så jag tolkar i alla fall det som att databasen är klar för normaliseringsform två. Gällande normaliseringsform tre så kan jag inte finna några transitiva beroenden till icke-nyckel attribut som villkoret är, och därför är det min tolkning att databasen är även klar för normaliseringsform tre. Möjligtvis skulle man kunna se det som så att en stad är beroende av det land de ligger i och i den här kontexten är landet sammanknutet med artisten, vilket inte heller stämmer eftersom primärnyckeln utgör av ett nummer. Men återigen enligt mina tolkningar så anser jag inte så vara fallet.

Normaliseringsformer är något som verkligen inte är lätt och kan orsaka mycket huvudbry så det är ändå med viss reservation jag klargör dessa normaliseringsformer och de är utifrån min tolkning. Om det ändå mot förmodan skulle vara så att det visar sig enligt en annan tolkning att någon normaliseringsform ändå inte skulle vara uppfylld så hänvisar jag då till att dels så fungerar databasen mycket bra och dels så blir användarvänligheten mycket bättre med att man anger som nämnts ovan både ursprungsland och stad. Man hade klarat sig med bara stad förvisso men det finner jag en aning okomplett.

Till sist så för de av Er läsare som inte alls vet vad Normaliseringsformer är så kan jag kort och extremt förenklat beskriva det som sätt att slippa redundant information i en databas. Varje enskild bit av information skall enbart finnas på ett enda ställe i en databas. Genom olika typer av beroenden, så kallade relationer, mellan tabeller i en databas så kan man upprätthålla detta även mellan tabeller. Därför som nämnts på andra ställen i denna dokumentation radera en artist före alla skivor där denne förekommer på är raderade. På omvänt sätt kan man inte heller därför lägga till en skiva med en artist som inte finns i artisttabellen. Med andra ord varje bit information får endast förekomma en gång i en och samma databas. Man kan förvisso ge tillåtelse till poster i en tabell att godkänna redundant data. Annars skulle man ju inte kunna ha flera skivor med samma artist. I och för sig blir ju posten om man ser en post som en hel enhet unik ändå på grund av att det är samma artist men just den skivan förekommer bara en gång, och då menar jag precis den skivan. Men för att göra livet än mer krångligt kan man även tillåta det, för att kunna öppna för möjligheten att ha två exemplar eller fler av samma skiva. Det kan man i sin tur frånga genom att ha ett fält för antal av den skivan. Som jag nämnt innan normaliseringsformer är något mycket komplext och innebär så mycket mer än det lilla jag nämnt här. För att få mer information och en bredare bild rekommenderar jag att söka efter information på Internet om dessa.

Optimering

Optimeringen för denna databas är i ett default-läge mycket löst satt. Det jag har utgått från är mina tidigare nämnda Accessdatabaser som för närvarande rymmer bland annat min skivdatabas. Jag har dock strävat efter att sätta så korrekta datatyper som möjligt och/eller lämpliga för ändamålet. Inmatningslängder och datastorlekar har jag anpassat efter hur det har sett ut i min Accessdatabas för skivor där det just nu lagras mellan 1000 och 1100 skivor vilket jag anser vara en relativt bra variationsgrund för att till exempel se vad lämplig maxlängd för en skivtitel kan vara. För att databasen skall vara skalbar har jag sedan lagt till extra utöver de värden jag kommit fram till, och för vissa fält kanske väl i överkant men det går att senare justera om man så vill.

Skall man vidare ha en optimering som är precis efter ens behov så behöver man ju göra ordentliga prestanda-tester under så realistiska former som möjligt. Det kräver rejäla resurser vilket jag av förklarliga skäl inte alls har tillgång till. Så skall man använda den här applikationen således i skarpt läge så krävs det som sagt dylika justeringar.

Den här versionen av applikationen är ju dessutom försedd med ett cache-objekt för ökad prestanda vilket även det i ett skarpt läge kräver justeringar i enlighet med resultaten i given kontext vid prestandatest och dylikt ger. Det jag bidrager med i sammanhanget är än en gång ett slags default-läge att utgå ifrån. Men ju mer krävande miljö med många användare, stora sökresultat och så vidare desto mer vitala blir de här frågorna.

Applikationslogik

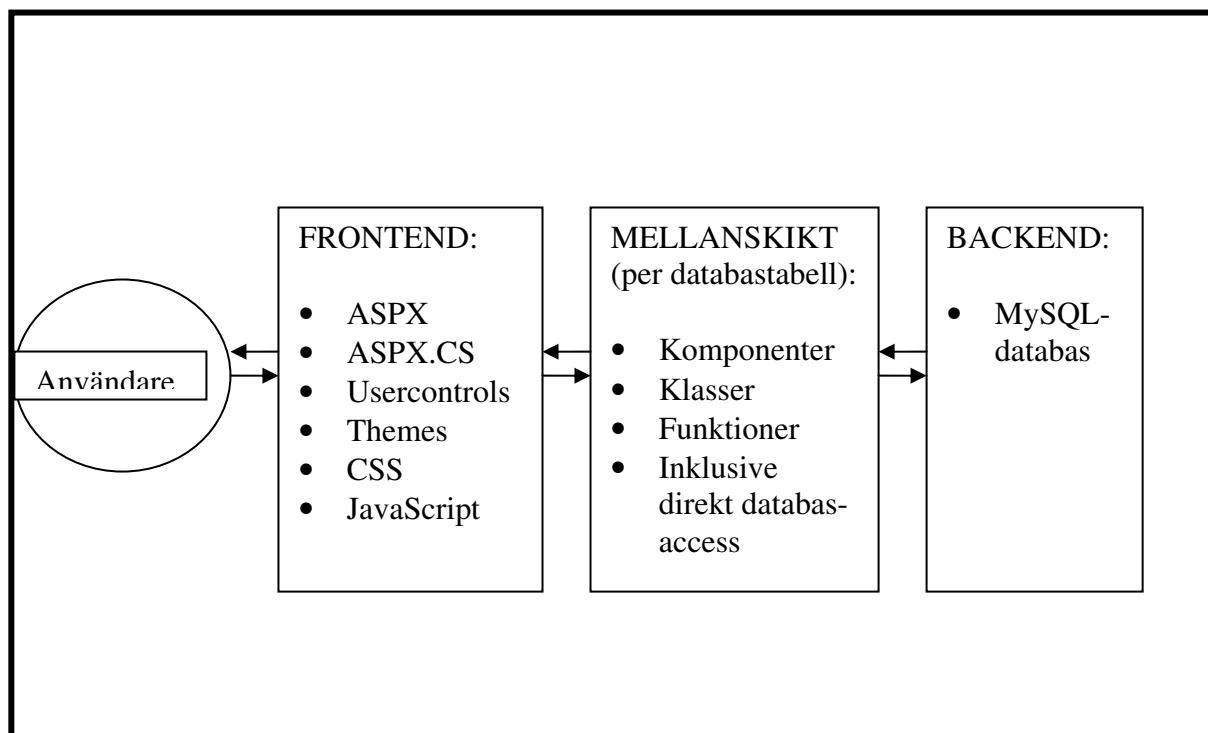


Bild2: Schematiskt förenklad bild över applikationslogiken.

Applikationslogiken för det här programmet tar sin startpunkt i den s.k treskiktsarkitekturen (three-tier). De tre skikten för min applikation har jag valt att kalla för frontend, mellanskipt, och backend (förklaring om detta följer längre fram i dokumentationen).

En annan viktig bakomliggande tanke/begrepp som är central för denna applikation är modularitet samt därmed komponentbaserad utveckling. Det har varit min strävan att särskilja så många ingående delar av denna applikation som möjligt, även i frontend-skiktet som är själva webbapplikationen och därmed inte ingår några komponenter men där jag som ni kan läsa om nedan ändå försökt applicera samma tankesätt. Fördelen med mer modulära applikationer är att man kan forma dem som man vill, kan lättare välja vilka delar som man vill använda och vilka man inte vill använda. Den uppsättning som jag har just nu går med andra ord lätt ändra med vissa justeringar i koden. Har man å andra sidan ett system som är mer integrerat och allt så att säga under samma skal, så kan man vinna vissa fördelar med en än högre grad av exempelvis dynamisk kod. Här i min applikation så har jag kodat till bristningsgränsen just dynamiskt men det modulära har ändå fått prioritet vilket kan ses i många fall som väl redundant, det vill säga snarlik, om än inte precis lika kod, på flertalet ställen. Men detta är just för att man skall kunna med enkla justeringar få den sammansättning av funktioner som man vill. Det är inte i lika hög grad möjligt i det andra nämnda fallet.

Frontend är det lager som ligger närmast användaren och dennes gränssnitt i form av knappar och inmatningsrutor i webbgrenssnittet. Där kan denne mata in värden i inmatningsrutor för att sedan genom att klicka på knapparna adressera värdena till en viss typ av händelse, då den så kallade bakomliggande koden i cs-filer reagerar med hjälp av händelsehanterare på just

nämnda och andra händelser. Programmet kan i sin tur meddela användaren om resultaten av händelsen i form av exempelvis en utskrift på displayen från en databassökning (men observera att det tillhör mellanskiktet, frontend-skiktet skriver bara ut felmeddelanden). Programmet kan även om ett fel uppstår skriva ut aktuellt felmeddelande på skärmen. Felmeddelanden sänds från de två övriga skikten vidare till detta första skikt för utskrift till användare. Detta sker genom så kallade try-catch satser. Detta första skikt anger som argument i sina anrop objekt som exempelvis olika webbkontroller eller Rich controls som gridviews. Kort sagt allt som kan behövas för hantering i mellanskiktet.

Ytterligare en sak som man bör nämna gällande struktur och modularitet gällande detta första skikt är användandet av så kallade rich controls i form av användarkontroller (user controls) som i mitt fall är gridviews, som är en slags avancerade tabeller, och paneler.

Användarkontroller har beträffande hantering stor likhet med det som i ASP gick under begreppet inserts, dvs kodavsnitt som låg i en extern fil och kunde på valfri plats i markup-dokumentet (det som numera är aspx-filer) sättas in genom att med en liten kodsnudd ange detta och adressera till aktuell insert-fil och dess innehåll. Detta gav en modularitet som jag blev mycket förtjust i. Därför använder jag mig av användarkontroller som jag tolkar som dess motsvarighet i ASP.NET. Men där är det hela objekt med tillhörande cs-filer med bakomliggande kod. Det vill säga både en fil som markerar upp själva strukturen och en tillhörande fil i par som ger utökade funktionalitet, precis som för aspx och aspx.cs-filer. Detta gör att man kan särskilja dem precis som med komponenter då varje gridview är en enskild fil (eg par med två filer som sagt) och likadant med paneler. Det gör även att struktur och funktionalitet särskiljs, vilket i sin tur underlättar exempelvis administration eller expansion. Vinsterna med detta är samma modularitet som jag eftersträvat som en röd tråd genom applikationen. Det är lättare att byta ut, modifiera och så vidare enligt behov. Eller för att uttrycka det bättre gällande alla sådana vinster i min applikation; Skalbarheten ökar, liksom smidigheten när behoven och smaken växlar.

Mellanskiktet, för att övergå till detta, består av en mängd komponenter, tio stycken för att vara exakt. Dessa i sin tur består av klasser som i sin tur består av funktioner för diverse göromål mot de två andra skikten, och kanske främst frontend-skiktet. Det finns bland annat tre stycken komponenter för just databasfunktionaliteter som lägga till och radera i databas, en för varje tabell i databasen. Vilket ökar den nämnda modulariteten då man både kan välja vilka tabeller man vill ha funktionalitet för, samt inom varje komponent välja vilka klasser och därmed vilka funktioner som passar för ens behov. Är man än mer händig så kan man även se dessa som mallar och lägga till nya klasser och kanske t.o.m komponenter. Men jag tror dessa funktionaliteter skall täcka de flesta behov.

Ser vi sedan vidare till hur en typisk klass i en av dessa komponenter fungerar så tar en sådan som första åtgärd emot anrop och nyttjar parametrar från frontend-skiktet i sin behandling i aktuell funktion. I en sådan funktion så ingår i princip all behandling förutom att ta emot och hämta själva datan från databasen. Den når man access till genom anrop till funktioner för detta hos andra komponenter avsedda för just det syftet. Även dessa är tre till antalet och uppdelade efter de tre tabeller som nyttjas. Så man kan med andra ord precis på samma sätt som för komponenterna som tar emot anropet direkt från frontend-skiktet, välja för vilka tabeller man vill ha access till databasen samt även vilka funktioner som man kan tänka sig nyttja. Även här tror jag de som jag har implementerat per default skall vara mer än nog. Och om inte så finns det som sagt var utrymme för ändringar.

Mellanskiktet tar även emot returvärden från de komponenter som har access till databasen, dessa kan sedan prövas, genom exempelvis se om några rader i databasen blivit påverkade, om de har blivit det så är det en lyckad process och en utskrift med de värden och dylikt som skall göras sker till display. Om inte så skrivs ett felmeddelande till displayen att så inte kunde ske. Endast enkla felmeddelanden av detta slag kan äga rum i mellanskiktet. Resterande, som har nämnts, sänds vidare till catch-satser i frontendskiktet för vidare hantering och utskrift av aktuellt felmeddelande till användare. Detta gäller även för de komponenter som har access till databasen.

Finns ytterligare tre komponenter i mellanskiktet som även de är uppdelade per tabell och vidare i funktioner etc, precis som de ovan nämnda. Dessa tar hand om vad jag har valt att kalla "vanliga uppgifter" (common tasks). Just nu i skrivande stund för admin-delen av applikationen finns det bara två klasser per komponent med en eller fler funktioner i varje klass. Den ena klassen rensar displayen från gridviews, paneler, utskrifter och dylikt som kan vara en lämplig åtgärd innan till exempel en ny gridview eller utskrift sker. Den här applikationen är för övrigt mycket uppbyggd på den logiken att visa och gömma olika objekt, utskrifter etcetera. Detta sker lite mer tekniskt genom ett anrop från frontend-skiktet i samband med en händelse som tex att en sökning genomförs. Då kan först ett anrop till aktuell komponent och funktion för den tabell som för tillfället nyttjas ske. Funktionen i den komponenten rensar då rent innan sökresultatet i en gridviewkontroll visas. Rent konkret sker det att man i anropet inkluderar aktuellt sid-objekt (page) som argument till motsvarande funktionsparameter i för tabellen aktuell motsvarande komponent, så att rensnings sker på just den sidan där anropet sker ifrån. Det är ett litet exempel på den form av dynamiskt tänkande och därmed programmering som är genomgående för den här applikationen. Studera gärna kodlistningen i detta dokument för att själva upptäcka fler exempel på detta.

Den andra klassen i dessa komponenter tar hand om kryssrutor i gridviews. Närmare bestämt så har även dessa access direkt till databas och ser till att de kryssrutor som är ikryssade på gränssnittet i frontend-skiktet även är det för motsvarande post i databasen (egentligen ändras ett numeriskt värde i databasen från noll till ett). Detta är viktigt för annars överensstämmer inte det som syns i gränssnittet med vad som verkligen sker i databasen.

Det finns till sist beträffande komponenter ännu en komponent som innehåller en förlängd/modulerad (egentligen ärvd) objekt till kryssrutan. Det som har lagts till är en egenskap som möjliggör lagring av ett värde i textformat för lättare identifiering av just aktuell kryssruta i kod.

Tanken i förlängningen för denna komponent är att man skall kunna addera ytterligare klasser med funktioner/er där man modulerar befintliga objekt till att mer passa specifika behov. Ytterligare en åtgärd i det dynamiska/modulära tänkandet som är den mest röda tråden av alla för den här applikationen.

Slutligen och till allra sist så har vi således i enlighet med treskiktsarkitekturen det skikt som ligger närmast den efterfrågade resursen, i det här fallet en MySQL-databas. Beskrivningen av denna samt några ord om hur den fungerar i just min applikation skall jag inte göra här då det har skett under ett antal andra avsnitt i denna dokumentation. Dessa är:

- Installation-databas
- Databasstruktur

- Normaliseringsformer
- Optimering
- och även under rubriken "I användning"

...jag hänvisar därför Er till dessa avsnitt istället.

ASP.NET-specifika funktionaliteter

Nedan så skall jag lista samt nämna lite om de mer specifika ASP.NET-funktionaliteter som ingår i min applikation (bitvis även gällande för client-delen också fastän det inte tas upp i dess dokumentation). Övergången från ASP till ASP.NET var ju minst ett stort steg som rent konkret har inneburit födseln av en rad nya tekniker. Därav tänkte jag att en dokumentation utan att nämna åtminstone merparten av dem som ingår i min applikation, inte skulle göra densamma fullständig.

Vissa av dessa är redan nämnda eller beskrivna på andra ställen i den här dokumentationen och kommer därför inte här beskrivas närmare.

- **Themes**

Kan ses som ett komplement till CSS-filer men med funktionaliteter som inte dessa uppfyller. Att styra utseende, layout och egenskaper för gridviews är något som jag har dem till i den här applikationen. Och då sådana således som inte går att kontrollera med hjälp av CSS-filer. Jag använder dem även i viss mån till labels.

- **Usercontrols (gridviews, panels)**

Beskrivs på andra ställen i denna dokumentation.

- **Web controls**

Web kontroller är namnet på alla de knappar, textrutor, etiketter med flera objekt som används på webbsidor för att utforma dessa. I och med att även de är just objekt och tillhör en klass så är ofta funktioner, en hel del egenskaper man kan ändra och så vidare, knutna till dem vilket gör dem mycket mer flexibla och kraftfulla än sina motsvarigheter i ASP för att inte tala om HTML.

- **State management (viewstate, session)**

State management består i något enklare uttryck att bevara en applikation eller delar av densamma i ett visst läge/status (state) och sedan kunna återkalla det vid behov. Som exempel så bevarar jag vid min custom paging (se nedan) i gridviews markerade poster så att de även är markerade när man lämnar aktuell sida och senare återkommer till denna. En annat vanligt användningsområde är att efter en sida har uppdaterats så finns när man återkomma till den värden i till exempel textrutor kvar. Innan dessa tekniker fanns fick man fylla i dem igen vilket bidrog till ett onödigt moment. Dessa tekniker har olika för- och nackdelar bland annat hur länge eller inom vilken räckvidd (scope) man kan spara värden i minnet. Viewstate har exempelvis räckvidd för endast den aktuella sidan medan Session sträcker sig till hela applikationen.

- **Error handling**

Beskrivs på andra ställen i denna dokumentation.

- **Validation**

Validering innebär att man utvärderar inmatad data och om den uppfyller vissa ställda kriterier så är den okej, och vice versa. I denna och även client-delen av applikationen så förekommer det hos vissa inmatningsrutor validering, som exempelvis att man bara kan mata in siffror eller bara mata in bokstäver och liknande tecken. Detta bidrager till att minska möjligheten att format-fel

uppkommer, vilket jag om de ändå skulle göra så har garderat mig mot med motsvarande felhantering.

- **Rich Controls (gridviews, panels)**

Beskrivs på andra ställen i denna dokumentation. Gridviews kanske man ändå skulle nämna några ord om vad det är för säkerhets skull. Det är kan man enkelt säga avancerade tabeller, där man kan utforma det mesta enligt tycke och smak på en rad olika sätt, även estetiskt och inte bara funktionellt. Exempelvis så har jag i mina gridviews sortering av många fält, där denna sker genom att klicka på en länkrubrik ovanför aktuellt fält (se även nedan gällande sorting).

- **ADO.NET (MySQL-provider: MySql.Data.MySqlClient)**

Den teknik som .NET-applikationer använder för att ansluta till en databas. Den består av två typer av grundläggande klasser. En för att spara och underhålla data, som till exempel DataTable eller DataSet. Den andra typen av klasser har hand om anslutningen till databaser eller för all del andra datakällor. Dessa brukar kallas för ADO.NET data providers, och existerar i många varianter. Varje sådan uppsättning av en variant motsvarar en viss typ av anslutning för exempelvis en viss typ av databas. Finns till exempel specifikt för Oracle-databas, MS-SQL-databas (default), ODBC eller som i mitt fall MySQL. Det kan, som i mitt fall, för att få det att fungera kräva installation av en speciell drivrutin då det kommer från den enorma flora av tredjepartstillverkare som förekommer och bidrager till den mängd datakällor som kan anslutas.

- **Component-Based Programming**

Beskrivs på andra ställen i denna dokumentation.

- **Caching**

Caching i min databas nyttjas till två ändamål: visning av en hel tabells innehåll samt för sökresultat då även dessa kan bli stora och belastningen därmed ökas på databasen, i synnerhet när det är många användare.

Cachingen i just denna admin-del kanske inte fyller lika stort praktiskt syfte som i klient-delen. Då det vid administration oftast är just administratören som använder applikationen och inte så många andra. Dock kan ju ett möjligt scenario för en administratör vara att man använder samma data ofta, och då kan ju cache-objektet komma till nytta även för en ensam administratör. På grund av det har jag satt cache-tiden, det vill säga tiden som det s.k cache-objektet sparar aktuell information i minnet innan det frigörs, till ynka 20 sekunder. Då blir det inte ett så störande moment om man inte vill nyttja det. Man kan både sätta det till noll eller om man vill vara något mer radikal ta bort hela den funktionen, vilket jag i och för sig skulle tycka var synd.

Cachingen däremot i client-delen har ett mycket mer klart användningsområde då det kan vara många användare som efterfrågar samma data, i synnerhet kanske för visning av hel tabell. Det ökar även prestandan för applikationen då frekvensen av förfrågningar direkt till databas minskas, vilket i sin tur bidrager till ökad skalbarhet om sådan önskas. Cache-tiderna är per default en timma för visning av hel tabell och tjugo minuter för sökresultat. Dessa kan enkelt justeras för andra värden.

- **Custom gridviews with custom paging and sorting**

Grunden i det här är att man ansluter sig programmatisk vid behov till datakällan

(datasource) som i min applikation är mysql-databasen eller närmare bestämt den av dess tre tabeller som man för tillfället skall nyttja. Man anropar från mellanskiktet motsvarande komponent, även denna i mellanskiktet, som har direkt access mot databasen och som retur får man ett så kallat dataset, som kan rymma en mängd tabeller om det är så. Men i det här fallet så är det vid exempelvis ett sökresultat de poster som kriterierna i söksträngen (sql-satsen) har motsvarat. Dessa sätts sedan som just datakälla till en gridview som är avsedd att visa sökresultat och när man i koden beslutar att denna gridviews synlighets (visible)-egenskap skall sättas till värdet sann så syns gridviewen i fråga på displayen och även så dess sökresultat om allt gått vägen. Kort sagt det här sker programmatisk eller dynamiskt om man så vill. Jag har även implementerat motsvarigheten att en gridview är ständigt kopplad mot en och samma funktion i mitt fall en select-funktion för att välja visa hela databasen. Då objektet som sköter detta (ObjectDataSource) bara kan vara uppkopplat mot en select-funktion per objekt så blir det mycket begränsat och anledningen till att jag inte använder det till mer än visa hela tabeller. Det blir till och med omöjligt att få till all den funktionalitet som krävs för en applikation som den här då det behövs i sanning mer än en select-funktion eller ja fler av samma sorts funktioner för att få ihop det hela. För övrigt en av de stora bristerna med denna typ av objekt i dagsläget.

När denna uppkoppling sker programmatiskt så följer även att man måste själv programmera så att funktioner som annars fungerar om man väljer den andra metodiken. Till dessa hör förmågan att kunna bläddra genom sökresultat och dylikt (paging) samt ovan nämnda sorteringsmöjlighet direkt via en titellänk på de poster i aktuell gridview där man önskar sådan funktionalitet (sorting). Jag skall inte här gå in på närmare exakt hur man gör det utan hänvisar till kodlistningen för det men kan säga att det var inte den enklaste uppgiften i den här världen att åstadkomma.

Slutligen så hör även gällande denna admin-del till att bevara som sagt state vid paging och filtreringar. Denna del är dock tyvärr något som jag inte i alla lägen fått till fullt ut beträffande kryssrutor och färgmarkeringar för poster. Men å andra sidan är det faktiskt det enda av alla funktionalitet för denna applikation, och även klient-delen, som inte lyckats helt. Det som fattats i skrivande stund gällande det och annat har jag satt under avsnittet "framtida eventuella förbättringar" lite längre fram i dokumentationen.

Inmatningsfält

Artist	Titel	Format	Press	År	Kommentar				
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>				
VPfr	VPtill	Ursland	Ursstad	Musiktyp	Kortbio	Inm_dat	Kgd	Kkat	CPris
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Bild3: Diverse inmatningsfält som används för att förse databasen med data.

Inledning

I och med att den här databasen bygger på tre stycken tabeller så för att mata in värden genom dess inmatningsfält så får man göra detta med hjälp av vissa kombinationer av rutor.

Alternativet hade varit än mer inmatningsfält, men det är redan många som det är. Därför så anger jag här nedan vilka fält som tillhör vilken tabell, och observera än en gång en del fält/rutor kan användas till mer än en tabell. Tabell väljer man med radioknapparna (se nedan under 'Knappar-funktionalitet'):

Cd/Skivtabellen:

Artist, Titel, Format, Press, År, Kommentar. Tänk även på att en artist matas in med artistnummer, och de finner man i Artisttabellen.

Artisttabellen:

Artist, VPfr, VPtill, Ursland, Ursstad, Musiktyp, Kortbio. Här går det dock bra till skillnad mot de två andra tabellerna att mata in en artist med bokstäver.

Inköpstabellen:

Artist, Titel, Format, Press, År, Kommentar, Inm_dat, Kgd, Kkat, Cpris. Tänk även på att en artist matas in med artistnummer, och de finner man i Artisttabellen.

Nedan följer även en kort beskrivning av varje inmatningsfält.

Artist

Här matas artistens nummer in eller för artisttabellen artistens namn direkt med bokstäver (se förklaring ovan).

Titel

Här matas den aktuella skivans titel in.

Format

Här väljs ett av format från drop-down menyn.

Press

Här väljs från drop-down menyn landet som skivan pressades i.

År

Här väljs från drop-down menyn det år som skivan gavs ut.

Kommentar

Här kan en kommentar om skivan ges. Max antal tecken som man kan använda inklusive blanktecken är 1000 stycken.

Vpfr (Verksam Period från)

Här väljs det för **Artisttabellen** årtal när artisten började bli verksam.

VPtill (Verksam Period till)

Här väljs det för **Artisttabellen** årtal när artisten slutade att vara verksam. Om denna fortfarande är verksam så väljs alternativet active.

Ursland (Ursprungsland)

Här väljer man för **Artisttabellen** landsbeteckningen för det land som artisten bodde eller eventuellt var verksam i när denna började med musik alternativt slog genom.

Ursstad (Ursprungsstad)

Samma som för Ursland ovan men gällande motsvarande stad istället. Skrivs dock in manuellt.

Musiktyp

Som det låter, man väljer den musiktyp för **Artisttabellen** som motsvarar eller bäst motsvarar aktuell artists.

Kortbio (Kort biografi)

Här skriver man in för **Artisttabellen** en kort biografi. Max antal tecken som man kan använda inklusive blanktecken är 6000. Och ja, jag anser att en biografi på till och med på 6000 tecken är en kort sådan. En lång biografi mina vänner, det är en bok det!

För att få en snyggare layout i databasen och få med en länk till artistens hemsida samt inkludera en bild, så har jag valt att fylla i data enligt följande mall. Detta kan man dock göra som man vill och bara fylla i text men jag tycker det ser lite mer snitsigt ut. Mallen är enligt följande:

```
<html><body><br /><img class=dyn_pic src=bandbilder/[ARTISTBILD] /><br /><b>Hemsida:</b><br /><a href=http://[ARTIST-URL]>[ARTISTNAMN]</a><br /><br /><b>[ARTISTNAMN SOM RUBRIK]</b><br />
```

[SJÄLVA BIOGRAFIN]

Mellanrum mellan varje stycke i biografien mha:

Avslutning för biografidelen:

</body></html>

OBS!!!

All text måste hänga samman i en stor "klump", så inga halvtomma rader eller något liknande. Men det kan rekommenderas att första se biografien som den brukar ses, därefter fylla i med html och därefter som sagt "klumpa ihop" allting.

Inm_dat (Inmatningsdatum)

Här anger man i **Inkopstabellen** det datum när aktuell skiva/post matades in. Detta medför att man kan se vilka de senast inlagda skivorna är, och därmed kanske också de som man vill köpa in först.

Kgd (Köpgrad)

Här anger man i **Inkopstabellen** i vilken grad man vill köpa skivan på en skala från 1 till 5. Detta så att man skall bland annat kunna sortera ut de skivor man vill köpa in först.

Kkat (Köpkategori)

Här anger man i **Inkopstabellen** vilken kategori av inköp aktuell skiva tillhör. Detta medför sedan lämplig sortering efter kategori, vilket kan underlätta inköp.

CPris (Cirka pris)

Här anger man i **Inkopstabellen** ett ungefärligt pris för varje skiva. Detta gör det lättare att beräkna kostnad för inköp.

Knappar-funktionalitet

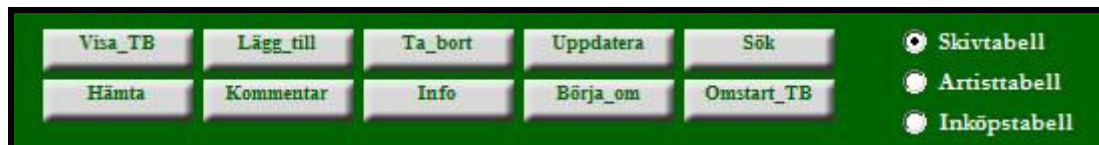


Bild4: Knappar som används för att kunna administrera och navigera databasen för Skivtabellen.



Bild5: Knappar som används för att kunna administrera och navigera databasen för Artisttabellen.



Bild6: Knappar som används för att kunna administrera och navigera databasen för Inköptabellen.

Inledning

Navigeringen i denna databas genomförs på så sätt att man först väljer vilken tabell man vill utföra åtgärd/inmatning på genom att välja en av radioknapparna till höger och därmed dess motsvarande tabell i databas och även som synes av skärmdumparna ovan knappsats. Dessa knappsatser är ganska lika men inte exakt därav att de är tre.

Nedan redogörs för hur var och en av knapparna fungerar för respektive knappsats och underförstått motsvarande tabell i databas.

Visa_TB

För Cd/Skivtabellen:

Visar alla köpta skivor som ingår i databasen. Är sorterade först efter artist och sedan efter albumtitel.

För Artisttabellen:

Visar alla artister som ingår i databasen, sorteras efter artistnamn.

För Inköptabellen:

Visar alla skivor som är tänkta att köpas in till databasen, sorteras först efter artist och sedan efter albumtitel.

Lägg_till

För Cd/Skivtabellen:

Lägger till en post i databasen. Tänk på att först i Artisttabellen kolla vad den artist som skall läggas in har för artistnummer och sedan använd detta. Artisnamn skall inte läggas in i klartext, detta är en säkerhetsspärr så att databasens integritet kvarstår och för att förhindra att redundant data matas in eller artister som inte finns i Artisttabellen. Det förhindrar även felstavning o.dyl av artister. **Artistfältet valideras för korrekt inmatning.**

För Artisttabellen:

Lägger till en ny artist. I Artisttabellen kan man dock lägga in artister direkt med bokstäver. **Artist- och Ursstadfälten valideras för korrekt inmatning.**

För Inköpstabellen:

Lägger till en post i databasen. Tänk på att först i Artisttabellen kolla vad den artist som skall läggas in har för artistnummer och sedan använd detta. Artisnamn skall inte läggas in i klartext, detta är en säkerhetsspärr så att databasens integritet kvarstår och för att förhindra att redundant data matas in eller artister som inte finns i Artisttabellen. Det förhindrar även felstavning o.dyl av artister. **Artist-, Inm_dat- och Cpris-fälten valideras för korrekt inmatning.**

Ta_bort

För Cd/Skivtabellen:

Tar bort en eller flera post/er i tabellen. Den eller de poster som skall tas bort väljs genom att markera motsvarande kryssruta för den eller de posterna.

För Artisttabellen:

Samma som för Cd/skiv-tabellen.

För Inköpstabellen:

Samma som för Cd/skiv-tabellen.

Uppdatera

För Cd/Skivtabellen:

Uppdaterar (ändrar) innehållet för en post eller delar av den. Måste föregås av en Hämta-funktion.

För Artisttabellen:

Samma som för Cd/skiv-tabellen.

För Inköpstabellen:

Samma som för Cd/skiv-tabellen.

Sök

För Cd/Skivtabellen:

Genomför en sökning. Sökning kan ske utifrån alla inmatningsfält för Skivtabellen, dvs utifrån Artist, Titel, Format Press, År samt Kommentar och vilka kombinationer som helst av dessa. I de fält där fri text kan matas in så kan man söka på delar av ord som t.e.x 'kr' för Kraftwerk gällande artist. Känner heller ingen skillnad mellan stora och små bokstäver. Sökresultaten är sorterade efter albumtitel.

För Artisttabellen:

Genomför en sökning. Sökning kan ske utifrån alla inmatningsfält för Artisttabellen, dvs utifrån Artist, VPfr, VPtill, Ursland, Ursstad, Musiktyp samt Kortbio och vilka kombinationer som helst av dessa. I de fält där fri text kan matas in så kan man söka på delar av ord som tex 'kr' för Kraftwerk gällande artist. Känner heller ingen skillnad mellan stora och små bokstäver. Sökresultaten är sorterade efter artistnamn.

För Inköpstabellen:

Genomför en sökning. Sökning kan ske utifrån alla inmatningsfält för Skivtabellen, dvs utifrån Artist, Titel, Format, Press, År, Kommentar, Inm_dat, Kgd, Kkat samt Cpris och vilka kombinationer som helst av dessa. I de fält där fri text kan matas in så kan man söka på delar av ord som t.e.x 'kr' för Kraftwerk gällande artist. Känner heller ingen skillnad mellan stora och små bokstäver. Sökresultaten är sorterade efter albumtitel.

Hämta

För Cd/Skivtabellen:

Hämtar ner valda fält från den post vars kryssruta man har valt. Därefter kan man ändra på uppgifterna, och sedan uppdatera uppgifterna för aktuell tabell (se även 'Uppdatera' ovan). Endast en post i taget kan väljas för uppdatering.

För Artisttabellen:

Samma som för Cd/skiv-tabellen.

För Inköpstabellen:

Samma som för Cd/skiv-tabellen.

Kommentar

För Cd/Skivtabellen:

Visar en panel där man kan om man anger det kommentarnummer man vill se (erhålls genom att kolla upp en intressant post när hel tabell visas eller från ett sökresultat) kan granska den kommentaren på ett lättare sätt på en större yta än vid inmatning med hjälp av motsvarande inmatningsruta för kommentarer. Panelen kan för övrigt ses som ett komplement till denna. Vidare så finns det möjlighet att ändra innehållet i vald kommentar, eller skriva in en helt ny kommentar, och mata in denna i databasen och resultatet av inmatningen visas direkt, samt även en knapp för rensa innehållet i panelen och rutan där man matar in kommentarnummer.

För Artisttabellen:

Förkommer inte för Artisttabellen.

För Inköpstabellen:

Samma som för Cd/skiv-tabellen.

Kortbio

Korbioknappen finns **bara under Skivtabellen**. Denna har stora likheter med kommentarpanelen som finns för de andra två tabellerna förutom att den hanterar på motsvarande sätt artisters kortbio som man kan även mata in med hjälp av en inmatningsruta. Då även en kort biografi kan bli rejält lång för en inmatningsruta så blir hanteringen mycket bättre i en panel som den här. I likhet med kommentarpanelerna som sagt så kan man visa befintliga kommentarer, ändra i dem, mata in ändrade kommentarer eller helt nya samt rensa panelens display och inmatningsrutan för kortbionumren.

Info

Info finns **bara under Skivtabellen**. Så välj denna tabell med den översta radioknappen och klicka sedan på denna knapp för att få fram info om skivtabellens aktuella innehåll.

Visar information om innehållet i skivtabellen såsom hur många sorters skivor/media av olika slag som för tillfället lagras under Skivtabellen i databasen. Både del- och totalsummeringar av den här informationen förekommer på displayen för att på så sätt lättare ge en mer fullständig överblick.

Överför

Den här funktionen är unik då den omfattar två tabeller samtidigt. Den för över, eller mer korrekt uttryckt flyttar, vald post från **Inkopstabellen** till **Skivtabellen**. Tanken är att efter man köpt en skiva som står angiven i Inkopstabellen så skall den kunna flyttas över från denna till Skivtabellen. Denna funktion gör just det åt dig.

Börja_om

Rensar alla inmatningsfält från det som eventuellt står i dem. **Finns för samtliga tabeller.**

Omstart_TB

Rensar alla inmatningsfält samt nollställer displayen. **Finns för samtliga tabeller.**

Holder

Holder finns **bara under Artisttabellen**. Det är precis som namnet antyder en platshållare för en eventuell framtida knapp med tillhörande funktionalitet. Den händer alltså inget för närvarande om man trycker på denna.

Webbläsare och upplösningar

Den här webbdatabasen är anpassad till att fungera med följande webbläsare: Mozilla Firefox 3.0, Google Chrome 3.0 samt Internet Explorer 7.0.

Beträffande upplösningar så har den testats och skall fungera i följande upplösningar:

1024*768
1152*864
1280*768
1280*800
1280*1024
1440*900
1680*1050

Dock rekommenderas att den inte används i lägre upplösningar än 1024*768 då gränssnittet blir alltför stort och otympligt med scrollbars etcetera. Högre upplösningar än ovan har jag inte haft möjlighet att testa så det kan jag inte uttala mig om men ser inget rent teoretiskt hinder i alla fall varför det inte skulle fungera, förutom möjligtvis att alla detaljer blir mycket, mycket små. I så fall så kan man skämtsamt säga att förstoringsglas rekommenderas.

I användning

Denna webbdatabas kan man nästan i princip använda hur man vill men jag har i alla fall tänkt mig att den skall användas som så att Admin-applikationen finns lokalt installerad på en dator och exekveras med hjälp av en lokal server. Men man ansluter till en MySQL-databas som ligger online på Internet eller vilken databas man nu vill använda (eftersom jag har använt mig av en för detta arbetsprov fortsätter jag på den linjen). På det här sättet kan man nå och administrera den aktuella databasen t.e.x från sitt eget hem.

Klientapplikationen skall däremot, och liksom ovan nämnda MySQL-databas, finnas online på nätet och kanske lämpligtvis på samma server som MySQL-databasen även om det självklart inte är något krav bara applikationen kan ansluta korrekt mot denna. Därmed kan alla som vill titta i webbdatabasen gå till aktuell webbadress för klientapplikationen, och den blir därmed nåbar globalt om man så vill.

Denna konfiguration kan skapa förvirring i och med att när jag nu visar upp denna databas-applikation i "arbetsprov-format" så ligger både administrationsdelen likväl som klientdelen online. Detta är för att dels kunna visa upp båda applikationerna som utgör databasen men dels också för att påvisa en aning att om man vill så kan även administrationsapplikationen ligga på Internet. Detta är som jag nämnt ovan inte min personliga grundtanke men det är fullt möjligt om man så önskar. Vidare har jag för just detta arbetsprov gällande administrationsdelen frångått funktionaliteten beträffande säker krypterad inloggning därför en sådan kan verka besvärande och tidskrävande för de som skall granska mitt arbetsprov. Finns dock upptaget i avsnittet "Framtida eventuella uppdateringar" som en möjlig framtida finess.

Framtida eventuella uppdateringar

Några funderingar på åtgärder gällande webbdatabasen som jag hoppas kan tillkomma i framtiden i mån av tid är följande:

- Ett login-skal/förstasida som omger administrations-applikationen så att den även kan ligga säkert online på Internet och nås därifrån. Login-funktionen skall vara försedd med kryptering för administratörernas, vilket är lämpligt. Dock kan den även självklart redan nu nyttjas på Internet och man kan förslagsvis sätta accessrättigheter till applikationen på den server som agerar host för den, så att bara vissa användare kan administrera den databas som den är kopplad mot. För detta arbetsprov så har jag frångått denna funktionalitet då en inloggning kan verka besvärande och tidskrävande för de som skall granska mitt arbetsprov.
- Kunna markera ALLA kryssrutor vid paging (gällande för gridviews som visar sökresultat). Inte bara som nu för aktuell sida i sökresultatet.
- Behålla färgmarkeringarna vid paging gällande gridviews som visar hela tabeller.
- I vissa fall så avmarkeras inte kryssrutor i gridviews efter uppdatering av aktuell sida, detta gällande vid paging och i synnerhet för de sidor som för tillfället inte är synliga.
- För bättrad valideringsfunktion (effektivare reguljärt uttryck) för inmatningsrutorna Artist.txt och Ursstad.txt. Denna skall filtrera bort alla tecken förutom bokstäver och mellanslag, dvs dessa skall vara de enda giltiga. Nuvarande reguljärt uttryck klarar inte av att ogiltigförklara följande tecken:

\$ ` ´ £ { } [] \ ^ ¨ ~ | . _

KÄLLKOD

Majoriteten av detta dokument (i alla fall räknat i antal sidor) består av den källkod som applikationen består av, en kodlistning om man så vill. Den är strukturerad först efter **frontend** och de filer som ingår där, därefter på liknande sätt **mellanskiktet (middletier)** och slutligen **backend**. Backend ingår egentligen inte direkt i min applikation då det inkluderar själva databasen och databashanteraren och så vidare. Men har ändå för att framställningen skall bli komplett nämnt några ord om databasen ändå.

Observera också att blandningen med begrepp från two-tier arkitektur och three—tier arkitektur är helt avsiktligt, då jag finner att backend och frontend är mer talande begrepp fastän de hör hemma i two-tier arkitektur och jag använder three-tier arkitektur. Därav denna blandning.

Nedan följer kort beskrivning av dessa tre begrepp, och på efterföljande sida börjar listningen av källkoden med kortfattade beskrivningar som motsvarar varje fil eller komponent.

Observera att konfigureringsfilen 'web.config' inte är med i listningen på grund av säkerhetsskäl.

Listningen av kod är kopierad från utvecklingsmiljön (Microsoft Visual Studio 2008) och där är inga specifika marginaler satta vilket gör att långa rader i utvecklingsmiljön blir brutna här och ser inte lika bra ut, utan kan ibland bli lite rörigt. Detta hade kunnat ändras men då hade man fått tagit bort den formatering som följer med från utvecklingsmiljön med bland annat färgmarkering vilket gör koden mycket mer lättläst. Jag valde att behålla formatering från utvecklingsmiljön vilket som sagt ibland kan göra att det inte ser så attraktivt ut här utan blir lite rörigt. Hoppas Ni har överseende med detta, då den ändå övergripande läsbarheten blir bättre än med föregående alternativ.

Frontend

"Begrepp som används inom informationsteknik för att beteckna den bearbetning som sker nära användaren (användargränssnittsorienterat) . I modern datorteknik associeras ofta Front-end med med webbbaserade mjukvarumoduler baserade på HTML och CSS "

(Wikipedia, URL: < http://sv.wikipedia.org/wiki/Front-end_och_back-end >, Senast besökt den 2009-12-31).

Mellanskikt (middle tier)

"This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers."

(upload.wikipedia.org, URL: < http://upload.wikimedia.org/wikipedia/commons/5/51/Overview_of_a_three-tier_application_vectorVersion.svg >, Senast besökt den 2009-12-31).

Backend

Kan man enkelt uttrycka är den del av en multi-skiktsapplikation som tar hand om slutdelen av de processer som förekommer. Den del som även är närmast den efterfrågade resursen som i mitt fall är en databas. Här nedan följer en något mer utförlig förklaring:

"A 'back-end' application or program serves indirectly in support of the front-end services, usually by being closer to the required resource or having the capability to communicate with the required resource. The back-end application may interact directly with the front-end or, perhaps more typically, is a program called from an intermediate program that mediates front-end and back-end activities."

(SearchDataCenter.com, URL: <

**http://searchdatacenter.techtarget.com/sDefinition/0,,sid80_gci212161,00.html
>, Senast besökt den 31 december 2009).**

FRONTEND

Frontend - Aspx i par med aspx.cs - filer

Markerar upp respektive sida med tillhörande bakomliggande kod som i sin tur innehåller anrop av funktionaliteter (i mellanskiktet).

Bildar strukturen för skivtabellen.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" ValidateRequest="false" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>SkivorNETAdmin</title>

<!-- JavaScript: -->
<script type="text/javascript" src="javascript/browser.js"></script>
<script type="text/javascript" src="javascript/datum.js"></script>

</head>
<body onload="datum(); setInterval('datum()', 1000 )">
    <form id="form1" runat="server">

        <div id="container">

            <div id="header">

                <!--Datumruta; Visning av Datum och tid: -->
                <div id="datumruta">

                    <div id="clock">&nbsp;</div>

                </div>

            </div>

            <div id="content">

                <!-- Display-ram börjar: -->
                <div class="ruta">

                    <!--Panel att "skriva" till: -->
                    <asp:Panel ID="frame" runat="server">

                        <!-- För att visa information/data som skall stå över
kontroll: -->
                        <span id="display" runat="server"
class="mess"></span>

                        <!-- GridView för att visa artistdata (i
usercontrol): -->
                        <TF:UGridResult ID="grdResult_UCtrl"
Visible="true" runat="server" />

                    </asp:Panel>

                </div>

            </div>

        </div>

    </form>

</body>
</html>
```

```

        <!-- GridView för att visa resultat från Hämta-
funktion och sökning (i usercontrol): -->
        <TF:UGridArtistdata ID="grd_Artistdata_UCtrl"
Visible="true" runat="server" />

        <!-- GridView som visar hel tabell inkl paging etc:
-->
        <TF:UGridArtister ID="gridArtister_Uctrl"
Visible="true" runat="server" />

        <!-- Panel för administrering av omfattande
kommentarer: -->
        <TF:UPnlKommentar ID="pnlKommentar_UCtrl"
Visible="true" runat="server" />

        <!-- För att visa information/data som skall stå
under kontroll: -->
        <span id="display2" runat="server"
class="mess"></span>

</asp:Panel>

        <!--Slut på display-ram: -->
</div>

</div>

<div id="navigation">

        <!--Labels RAD1: -->

        <asp:Label CssClass="label1 move" ID="label1_Artist"
runat="server" Text="Artist"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label2_Album"
runat="server" Text="Titel"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label3_Format"
runat="server" Text="Format"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label4_Press"
runat="server" Text="Press"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label5_Kommentar"
runat="server" Text="Kommentar"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label6_ar"
runat="server" Text="År"></asp:Label>

```

```

<!--Texttrutor RAD1: -->

    <asp:TextBox CssClass="move text1" ID="text3_Artist"
MaxLength="50"
        runat="server" Size="15"></asp:TextBox>

    <asp:RegularExpressionValidator id="vldArtistOnlyNum"
runat="server"
        Display="None" ValidationExpression="^[0-9]{0,4}$"
        ControlToValidate="text3_Artist" Enabled="true"
        EnableClientScript="false" ValidationGroup="x"/>

    <asp:RequiredFieldValidator ID="vldArtistNotEmpty"
runat="server"
        Display="None" ControlToValidate="text3_Artist"
Enabled="true"
        EnableClientScript="false" ValidationGroup="x"/>

    <asp:RangeValidator ID="vldArtistValidRange"
runat="server"
        Display="None" ControlToValidate="text3_Artist"
        Enabled="true" EnableClientScript="false"
Type="Integer"
        ValidationGroup="x"/>

    <asp:TextBox CssClass="move text1" ID="text4_Album"
MaxLength="80" runat="server" Size="15"></asp:TextBox>

    <asp:DropDownList CssClass="move text1" ID="text5_Format"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="CD" />
        <asp:ListItem Value="CDM" />
        <asp:ListItem Value="CDS" />
        <asp:ListItem Value="2CD" />
        <asp:ListItem Value="LCDM" />
        <asp:ListItem Value="XLCDM" />
        <asp:ListItem Value="DVDS" />
        <asp:ListItem Value="CD+DVD" />
        <asp:ListItem Value="DVD/SACD" />
        <asp:ListItem Value="CDM/DVD" />
        <asp:ListItem Value="dEP" />
        <asp:ListItem Value="LP" />
        <asp:ListItem Value="2LP" />
        <asp:ListItem Value="BOX" />
        <asp:ListItem Value="12" />
        <asp:ListItem Value="L12" />
        <asp:ListItem Value="XL12" />
        <asp:ListItem Value="L12num" />
        <asp:ListItem Value="TwinSetL12" />
        <asp:ListItem Value="TwinSetL12num" />
        <asp:ListItem Value="Promo" />
        <asp:ListItem Value="7" />
        <asp:ListItem Value="aEP" />
        <asp:ListItem Value="mp3" />
    </asp:DropDownList>

```



```

        <asp:DropDownList CssClass="move text1" ID="text6_Press"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="at" />
        <asp:ListItem Value="be" />
        <asp:ListItem Value="ca" />
        <asp:ListItem Value="de" />
        <asp:ListItem Value="dk" />
        <asp:ListItem Value="eu" />
        <asp:ListItem Value="fr" />
        <asp:ListItem Value="holl" />
        <asp:ListItem Value="it" />
        <asp:ListItem Value="se" />
        <asp:ListItem Value="uk" />
        <asp:ListItem Value="ukorg" />
        <asp:ListItem Value="us" />
    </asp:DropDownList>

```

```

        <asp:DropDownList CssClass="move text1" ID="text8_ar"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="1965" />
        <asp:ListItem Value="1966" />
        <asp:ListItem Value="1967" />
        <asp:ListItem Value="1968" />
        <asp:ListItem Value="1969" />
        <asp:ListItem Value="1970" />
        <asp:ListItem Value="1971" />
        <asp:ListItem Value="1972" />
        <asp:ListItem Value="1973" />
        <asp:ListItem Value="1974" />
        <asp:ListItem Value="1975" />
        <asp:ListItem Value="1976" />
        <asp:ListItem Value="1977" />
        <asp:ListItem Value="1978" />
        <asp:ListItem Value="1979" />
        <asp:ListItem Value="1980" />
        <asp:ListItem Value="1981" />
        <asp:ListItem Value="1982" />
        <asp:ListItem Value="1983" />
        <asp:ListItem Value="1984" />
        <asp:ListItem Value="1985" />
        <asp:ListItem Value="1986" />
        <asp:ListItem Value="1987" />
        <asp:ListItem Value="1988" />
        <asp:ListItem Value="1989" />
        <asp:ListItem Value="1990" />
        <asp:ListItem Value="1991" />
        <asp:ListItem Value="1992" />
        <asp:ListItem Value="1993" />
        <asp:ListItem Value="1994" />
        <asp:ListItem Value="1995" />
        <asp:ListItem Value="1996" />
        <asp:ListItem Value="1997" />
        <asp:ListItem Value="1998" />
        <asp:ListItem Value="1999" />
        <asp:ListItem Value="2000" />
        <asp:ListItem Value="2001" />
        <asp:ListItem Value="2002" />
        <asp:ListItem Value="2003" />

```

```

        <asp:ListItem Value="2004" />
        <asp:ListItem Value="2005" />
        <asp:ListItem Value="2006" />
        <asp:ListItem Value="2007" />
        <asp:ListItem Value="2008" />
        <asp:ListItem Value="2009" />
        <asp:ListItem Value="2010" />
        <asp:ListItem Value="2011" />
        <asp:ListItem Value="2012" />
        <asp:ListItem Value="2013" />
        <asp:ListItem Value="2014" />
        <asp:ListItem Value="2015" />
        <asp:ListItem Value="2016" />
        <asp:ListItem Value="2017" />
        <asp:ListItem Value="2018" />
        <asp:ListItem Value="2019" />
        <asp:ListItem Value="2020" />

    </asp:DropDownList>

    <asp:TextBox CssClass="move text1" ID="text7_Kommentar"
MaxLength="1000" runat="server" Size="15"></asp:TextBox>

<!--Labels RAD2: -->

    <asp:Label CssClass="label2 move" ID="label7_VPfr"
runat="server" Text="VPfr"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label8_VPtill"
runat="server" Text="VPtill"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label9_Ursland"
runat="server" Text="Ursland"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label10_Ursstad"
runat="server" Text="Ursstad"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label11_Musiktyp"
runat="server" Text="Musiktyp"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label12_Kortbio"
runat="server" Text="Kortbio"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label13_Inm_dat"
runat="server" Text="Inm_dat"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label14_Kop_grad"
runat="server" Text="Kgd"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label15_Kop_kat"
runat="server" Text="Kkat"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label16_Ca_pris"
runat="server" Text="CPris"></asp:Label>

<!--Texttrutor RAD2: -->

```

```

        <asp:DropDownList CssClass="move text2" ID="text9_VPfr"
runat="server" Size="1">
    <asp:ListItem />
    <asp:ListItem Value="1965" />
    <asp:ListItem Value="1966" />
    <asp:ListItem Value="1967" />
    <asp:ListItem Value="1968" />
    <asp:ListItem Value="1969" />
    <asp:ListItem Value="1970" />
    <asp:ListItem Value="1971" />
    <asp:ListItem Value="1972" />
    <asp:ListItem Value="1973" />
    <asp:ListItem Value="1974" />
    <asp:ListItem Value="1975" />
    <asp:ListItem Value="1976" />
    <asp:ListItem Value="1977" />
    <asp:ListItem Value="1978" />
    <asp:ListItem Value="1979" />
    <asp:ListItem Value="1980" />
    <asp:ListItem Value="1981" />
    <asp:ListItem Value="1982" />
    <asp:ListItem Value="1983" />
    <asp:ListItem Value="1984" />
    <asp:ListItem Value="1985" />
    <asp:ListItem Value="1986" />
    <asp:ListItem Value="1987" />
    <asp:ListItem Value="1988" />
    <asp:ListItem Value="1989" />
    <asp:ListItem Value="1990" />
    <asp:ListItem Value="1991" />
    <asp:ListItem Value="1992" />
    <asp:ListItem Value="1993" />
    <asp:ListItem Value="1994" />
    <asp:ListItem Value="1995" />
    <asp:ListItem Value="1996" />
    <asp:ListItem Value="1997" />
    <asp:ListItem Value="1998" />
    <asp:ListItem Value="1999" />
    <asp:ListItem Value="2000" />
    <asp:ListItem Value="2001" />
    <asp:ListItem Value="2002" />
    <asp:ListItem Value="2003" />
    <asp:ListItem Value="2004" />
    <asp:ListItem Value="2005" />
    <asp:ListItem Value="2006" />
    <asp:ListItem Value="2007" />
    <asp:ListItem Value="2008" />
    <asp:ListItem Value="2009" />
    <asp:ListItem Value="2010" />
    <asp:ListItem Value="2011" />
    <asp:ListItem Value="2012" />
    <asp:ListItem Value="2013" />
    <asp:ListItem Value="2014" />
    <asp:ListItem Value="2015" />
    <asp:ListItem Value="2016" />
    <asp:ListItem Value="2017" />
    <asp:ListItem Value="2018" />
    <asp:ListItem Value="2019" />
    <asp:ListItem Value="2020" />
</asp:DropDownList>

```

```

        <asp:DropDownList CssClass="move text2" ID="text10_VPtill"
runat="server" Size="1">
    <asp:ListItem />
    <asp:ListItem Value="active" />
    <asp:ListItem Value="1965" />
    <asp:ListItem Value="1966" />
    <asp:ListItem Value="1967" />
    <asp:ListItem Value="1968" />
    <asp:ListItem Value="1969" />
    <asp:ListItem Value="1970" />
    <asp:ListItem Value="1971" />
    <asp:ListItem Value="1972" />
    <asp:ListItem Value="1973" />
    <asp:ListItem Value="1974" />
    <asp:ListItem Value="1975" />
    <asp:ListItem Value="1976" />
    <asp:ListItem Value="1977" />
    <asp:ListItem Value="1978" />
    <asp:ListItem Value="1979" />
    <asp:ListItem Value="1980" />
    <asp:ListItem Value="1981" />
    <asp:ListItem Value="1982" />
    <asp:ListItem Value="1983" />
    <asp:ListItem Value="1984" />
    <asp:ListItem Value="1985" />
    <asp:ListItem Value="1986" />
    <asp:ListItem Value="1987" />
    <asp:ListItem Value="1988" />
    <asp:ListItem Value="1989" />
    <asp:ListItem Value="1990" />
    <asp:ListItem Value="1991" />
    <asp:ListItem Value="1992" />
    <asp:ListItem Value="1993" />
    <asp:ListItem Value="1994" />
    <asp:ListItem Value="1995" />
    <asp:ListItem Value="1996" />
    <asp:ListItem Value="1997" />
    <asp:ListItem Value="1998" />
    <asp:ListItem Value="1999" />
    <asp:ListItem Value="2000" />
    <asp:ListItem Value="2001" />
    <asp:ListItem Value="2002" />
    <asp:ListItem Value="2003" />
    <asp:ListItem Value="2004" />
    <asp:ListItem Value="2005" />
    <asp:ListItem Value="2006" />
    <asp:ListItem Value="2007" />
    <asp:ListItem Value="2008" />
    <asp:ListItem Value="2009" />
    <asp:ListItem Value="2010" />
    <asp:ListItem Value="2011" />
    <asp:ListItem Value="2012" />
    <asp:ListItem Value="2013" />
    <asp:ListItem Value="2014" />
    <asp:ListItem Value="2015" />
    <asp:ListItem Value="2016" />
    <asp:ListItem Value="2017" />
    <asp:ListItem Value="2018" />
    <asp:ListItem Value="2019" />
    <asp:ListItem Value="2020" />
</asp:DropDownList>

```

```

runat="server" <asp:DropDownList CssClass="move text2" ID="text11_Ursland"
Size="1">
  <asp:ListItem />
  <asp:ListItem Value="at" />
    <asp:ListItem Value="be" />
    <asp:ListItem Value="ca" />
    <asp:ListItem Value="de" />
    <asp:ListItem Value="dk" />
    <asp:ListItem Value="eu" />
    <asp:ListItem Value="fr" />
    <asp:ListItem Value="holl" />
    <asp:ListItem Value="it" />
    <asp:ListItem Value="se" />
    <asp:ListItem Value="uk" />
    <asp:ListItem Value="ukorg" />
    <asp:ListItem Value="us" />
  </asp:DropDownList>

  <asp:TextBox CssClass="move text2" ID="text12_Ursstad"
MaxLength="50" runat="server" Size="6"></asp:TextBox>

runat="server" <asp:DropDownList CssClass="move text2" ID="text13_Musiktyp"
Size="1">
  <asp:ListItem />
  <asp:ListItem Value="Cyberpunk" />
  <asp:ListItem Value="EBM" />
  <asp:ListItem Value="EuroDisco" />
  <asp:ListItem Value="Gothrock" />
  <asp:ListItem Value="HipHop" />
  <asp:ListItem Value="Industri" />
  <asp:ListItem Value="New Wave" />
  <asp:ListItem Value="Pop" />
  <asp:ListItem Value="Punk" />
  <asp:ListItem Value="RnB" />
  <asp:ListItem Value="Rock" />
  <asp:ListItem Value="Synth" />
  <asp:ListItem Value="Synthpop" />
</asp:DropDownList>

  <asp:TextBox CssClass="move text2" ID="text14_Kortbio"
MaxLength="6000" runat="server" Size="6"></asp:TextBox>

  <asp:TextBox CssClass="move text2" ID="text15_Inm_dat"
MaxLength="10"
    runat="server" Size="6"></asp:TextBox>

runat="server" <asp:DropDownList CssClass="move text2" ID="text16_Kop_grad"
Size="1">
  <asp:ListItem />
  <asp:ListItem Value="1" />
  <asp:ListItem Value="2" />
  <asp:ListItem Value="3" />
  <asp:ListItem Value="4" />
  <asp:ListItem Value="5" />
</asp:DropDownList>

runat="server" <asp:DropDownList CssClass="move text2" ID="text17_Kop_kat"
Size="1">
  <asp:ListItem />

```

```

        <asp:ListItem Value="Ny" />
        <asp:ListItem Value="Beg" />
        <asp:ListItem Value="Rea" />
    </asp:DropDownList>

    <asp:TextBox CssClass="move text2" ID="text18_Ca_pris"
MaxLength="4"
        runat="server" Size="1"
    ></asp:TextBox>

<!--Knappar RAD1: -->

<asp:Button ID="button_visa_databas" runat="server"
    CssClass="input move rad1" Text="Visa_TB"
    onclick="button_visa_databas_Click"
    CausesValidation="False" />

    <asp:Button ID="button_lagg_till" runat="server"
    CssClass="input move rad1" Text="Lägg_till"
    onclick="button_lagg_till_Click"
    CausesValidation="true" ValidationGroup="x"/>

    <asp:Button ID="button_ta_bort" runat="server"
    CssClass="input move rad1" Text="Ta_bort"
    onclick="button_ta_bort_Click"
    CausesValidation="False" />

    <asp:Button ID="button_uppdatera" runat="server"
    CssClass="input move rad1" Text="Uppdatera"
    onclick="button_uppdatera_Click"
    CausesValidation="False" />

    <asp:Button ID="button_sok" runat="server"
    CssClass="input move rad1" Text="Sök"
    onclick="button_sok_Click"
    CausesValidation="False" />

<!--Knappar - RAD2: -->

    <asp:Button ID="btn21" runat="server"
    CssClass="input move rad2" Text="Hämta"
    onclick="button_fetch_Click"
    CausesValidation="False"/>

    <asp:Button ID="btn22" runat="server"
    CssClass="input move rad2" Text="Kommentar"
    onclick="button_Kommentar_Click"
    CausesValidation="False"/>

    <asp:Button ID="btn23" runat="server"
    CssClass="input move rad2" Text="Info"
    onclick="button_info_Click"
    CausesValidation="False" />

```

```

        <asp:Button ID="btn24" runat="server"
        CssClass="input move rad2" Text="Börja_om"
        OnClientClick="this.form.reset();return false;"
        onclick="button_Borja_om_Click"
        CausesValidation="False" />

        <asp:Button ID="btn25" runat="server"
        CssClass="input move rad2" Text="Omstart_TB"
        onclick="button_Refresh_TB_Click"
        CausesValidation="False" />

<!-- Tabellvals-radioknappar inkl. labels (i form av div): -->

<div class="label3 move" id="label13_TBskivor">Skivtabell</div>

    <asp:RadioButton ID="optSkivtabell" runat="server"
    GroupName="radioval" TextAlign="Right"
    AutoPostBack="true" CssClass="optSkivtabell" Checked="true"
    OnCheckedChanged="Skivtabell_CheckedChanged" />

    <div class="label3 move"
    id="label14_TBartist">Artisttabell</div>
    <asp:RadioButton ID="optArtisttabell" runat="server"
    GroupName="radioval" TextAlign="Right"
    AutoPostBack="true" CssClass="optArtisttabell"
    OnCheckedChanged="Artisttabell_CheckedChanged"
    Checked="false" />

    <div class="label3 move" id="label15_TBinkop">Inköpstabell</div>
    <asp:RadioButton ID="optInkopstabell" runat="server"
    GroupName="radioval" TextAlign="Right"
    AutoPostBack="true" CssClass="optInkopstabell"
    OnCheckedChanged="Inkopstabell_CheckedChanged"
    Checked="false" />

</div>

</div>
</form>
</body>
</html>

```

Bakomliggande kod till strukturen för skivtabellen. Här är det anrop till olika komponenter innehållandes funktioner som lägg till, ta bort etc finns. Mesta av felhanteringen sköts här och felmeddelanden ges till användaren när sådana har uppkommit.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Collections;
using System.Collections.Specialized;
using System.Drawing;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;
using DB_Callcode.Skivor;

public partial class _Default : System.Web.UI.Page
{
    //Instans för rensning av display och gridviews
    Clear clr = new Clear();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            //Initisiering av range-värden för validering av
            //vid tillägg av enhet(Lagg_till funktionen):
            Proc_act aggz = new Proc_act();
            int[] aggvalues = new int[2];
            aggvalues = aggz.GetAggCDNet();

            int minvalue = aggvalues[0];
            int maxvalue = aggvalues[1];

            vldArtistValidRange.MinimumValue = minvalue.ToString();

            if (maxvalue <= int.MaxValue)
            {
                vldArtistValidRange.MaximumValue = maxvalue.ToString();
            }
        }
    }

    //Knappar Rad 1:

    //Hämtar hela databasen:
```



```

        public void button_visa_databas_Click(object sender, EventArgs
e)
        {
            //Leta upp kontrollern:
            UserControl gridArtister_Uctrl =
(UserControl)Page.FindControl("gridArtister_Uctrl");

            GridView gridArtister =
(GridView)gridArtister_Uctrl.FindControl("gridArtister");

            Visa_databas vdb = new Visa_databas();
            vdb.button_visa_databas(gridArtister, this.Page);
        }

        //Lägger till data i databas:
        protected void button_lagg_till_Click(object sender, EventArgs
e)
        {
            try
            {
                Lagg_till lt = new Lagg_till();
                lt.button_lagg_till(display, text3_Artist, text4_Album,
text5_Format, text6_Press, text8_ar, text7_Kommentar,
vldArtistOnlyNum, vldArtistNotEmpty,
vldArtistValidRange,
                this.Page);
            }
            catch (FormatException err)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(this.Page);

                string mess = "<h2>[Page]Skivor.Lagg_till:
FormatException</h2>";
                mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
                display.InnerHtml = mess;
            }

            catch (SQLException err)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(this.Page);

                string mess =
"<h2>[Page]Skivor.Lagg_till:SQLException</h2>";
                mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
                display.InnerHtml = mess;
            }

            catch (System.Exception err)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(this.Page);

                string mess =
"<h2>[Page]Skivor.Lagg_till:System.Exception</h2>";
            }
        }
    }
}

```

```

        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Tar bort data från databas:
protected void button_ta_bort_Click(object sender, EventArgs e)
{
    //Leta upp kontroller:
    GridView grdResult =
(GridView)grdResult_UCtrl.FindControl("grdResult");
    GridView gridArtister =
(GridView)gridArtister_Uctrl.FindControl("gridArtister");

    try
    {
        Ta_bort tb = new Ta_bort();
        tb.button_ta_bort(gridArtister, grdResult, display,
this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page]Skivor.Ta_bort:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page]Skivor.Ta_bort:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page]Skivor.Ta_bort:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Uppdaterar databas:

```

```

        protected void button_uppdatera_Click(object sender, EventArgs
e)
        {
            //Leta upp kontrollerr:
            GridView grdResult =
(GridView)grdResult_UCtrl.FindControl("grdResult");
            GridView gridArtister =
(GridView)gridArtister_Uctrl.FindControl("gridArtister");

            try
            {
                Uppdatera upd = new Uppdatera();
                upd.button_uppdatera(gridArtister, grdResult, display,
this.Page, text3_Artist, text4_Album, text5_Format,
text6_Press, text8_ar, text7_Kommentar);
            }
            catch (FormatException err)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(this.Page);

                string mess =
"<h2>[Page] Skivor.Uppdatera:FormatException</h2>";
                mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
                display.InnerHtml = mess;
            }
            catch (MySqlException err)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(this.Page);

                string mess =
"<h2>[Page] Skivor.Uppdatera:MySqlException</h2>";
                mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
                display.InnerHtml = mess;
            }
            catch (System.Exception err)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(this.Page);

                string mess =
"<h2>[Page] Skivor.Uppdatera:System.Exception</h2>";
                mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
                display.InnerHtml = mess;
            }
        }

        //Söker i databas:
        protected void button_sok_Click(object sender, EventArgs e)
        {
            //Leta upp kontrollerr:
            GridView grdResult =
(GridView)grdResult_UCtrl.FindControl("grdResult");

            //Skapar instans av objekt för delayvärde och sök-anrop

```

```

        Sok sk = new Sok();

        //Anger antal sekunder som sökningen skall vara i cache-
objektet:
        sk.Delay = 20;

        try
        {
            sk.button_sok(grdResult, this.Page, display,
text3_Artist,
            text4_Album, text5_Format, text6_Press, text8_ar,
            text7_Kommentar);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            string mess =
"<h2>[Page] Skivor.Sok:FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            string mess =
"<h2>[Page] Skivor.Sok:MySqlException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (System.Exception err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            string mess =
"<h2>[Page] Skivor.Sok:System.Exception</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
    }

    //Ger information om aktuell data i databas:
    protected void button_info_Click(object sender, EventArgs e)
    {
        try
        {
            Info info = new Info();
            info.button_info(display, display2, this.Page);
        }
        catch (FormatException err)
        {

```

```

        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Info:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Info:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Info:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

//Knappar Rad 2:

```

//Hämtar en rad från aktuell tabell som skall uppdateras:
protected void button_fetch_Click(object sender, EventArgs e)
{
    //Leta upp kontroller:
    GridView grdResult =
(GridView)grdResult_UCtrl.FindControl("grdResult");
    GridView gridArtister =
(GridView)gridArtister_Uctrl.FindControl("gridArtister");

    try
    {
        Fetch fch = new Fetch();
        fch.button_fetch(gridArtister, grdResult, display,
display2, this.Page, text3_Artist, text4_Album,
text5_Format,
        text6_Press, text8_ar, text7_Kommentar);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);
    }
}

```

```

        string mess =
"<h2>[Page] Skivor.Fetch:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Fetch:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Fetch:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

```

//Visar panel för inmatning av längre kommentarer till
skivtabell:
protected void button_Kommentar_Click(object sender, EventArgs
e)
{
    //Leta upp kontroller:
    Panel pnlKommentar =
(Panel)pnlKommentar_UCtrl.FindControl("pnlKommentar");

    try
    {
        Visa_KommentarPanel kpnl = new Visa_KommentarPanel();
        kpnl.button_kommentarpanel(pnlKommentar, this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Kommentar:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);
    }
}

```

```

        string mess =
"<h2>[Page] Skivor.Kommentar:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Kommentar:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

```

//Raderar innehållet i alla textrutor och dropdownmenyer:
protected void button_Borja_om_Click(object sender, EventArgs
e)
{
    Borja_om bro = new Borja_om();
    bro.button_Borja_om();
}

```

```

//Fungerar som en omstart och rensar alla inmatningsrutor odyl
+ display:
protected void button_Refresh_TB_Click(object sender, EventArgs
e)
{
    HttpServerUtility server = this.Server;
    String retursida = "~/Default.aspx";

    try
    {
        Refresh_TB rtb = new Refresh_TB();
        rtb.button_Refresh_TB(server, retursida);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Refresh_TB:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

```

    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Refresh_TB:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Page] Skivor.Refresh_TB:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

//Växla mellan tabeller:

```

protected void Skivtabell_CheckedChanged(object sender,
EventArgs e)
{
    IDictionaryEnumerator CacheEnum =
HttpRuntime.Cache.GetEnumerator();
    while (CacheEnum.MoveNext())
    {
        string key = CacheEnum.Key.ToString();
        HttpRuntime.Cache.Remove(key);
    }
    Server.Transfer("~/Default.aspx");
}

protected void Artisttabell_CheckedChanged(object sender,
EventArgs e)
{
    IDictionaryEnumerator CacheEnum =
HttpRuntime.Cache.GetEnumerator();
    while (CacheEnum.MoveNext())
    {
        string key = CacheEnum.Key.ToString();
    }
}

```



```

        HttpRuntime.Cache.Remove(key);
    }
    Server.Transfer("~/Default2.aspx");
}

protected void Inkoptabell_CheckedChanged(object sender,
EventArgs e)
{
    IDictionaryEnumerator CacheEnum =
HttpRuntime.Cache.GetEnumerator();
    while (CacheEnum.MoveNext())
    {
        string key = CacheEnum.Key.ToString();
        HttpRuntime.Cache.Remove(key);
    }
    Server.Transfer("~/Default3.aspx");
}

}

```

Bildar struktur för artisttabellen

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" validateRequest=false %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title>SkivorNETAdmin</title>

<!-- JavaScript: -->
<script type="text/javascript" src="javascript/browser.js"></script>
<script type="text/javascript" src="javascript/datum.js"></script>

</head>
<body onload="datum(); setInterval('datum()', 1000 )">
    <form id="form1" runat="server">

        <div id="container">

            <div id="header">

                <!--Datumruta; Visning av Datum och tid: -->
                <div id="datumruta">

                    <div id="clock">&nbsp;</div>

                </div>

            </div>

            <div id="content">

                <!-- Display-ram börjar: -->
                <div class="ruta">

                    <!--Panel att "skriva" till: -->
                    <asp:Panel ID="frame" runat="server">

                        <!-- För att visa information/data som skall stå över
kontroll: -->
                        <span id="display" runat="server"
class="mess"></span>

                        <!-- GridView för att visa artistdata (i
usercontrol): -->
                        <TF:UGridResult_Artist ID="grdResult_Artist_UCtrl"
Visible="true" runat="server" />

                    </asp:Panel>

                </div>

            </div>

        </div>

    </form>

</body>
</html>
```

```

                                <!-- GridView som visar hel tabell inkl paging
etc: -->
                                <TF:UGridArtister_Artist
ID="gridArtister_Artist_UCtrl" Visible="true" runat="server" />

                                <!-- Panel för in- och utmaning av artists korbio i
artst-tabell: -->
                                <TF:UPnlKortbio ID="pnlKortbio_UCtrl"
Visible="true" runat="server" />

                                <!-- För att visa information/data som skall stå under
kontroll: -->
                                <span id="display2" runat="server" class="mess"></span>

                                </asp:Panel>

                                <!--Slut på display-ram: -->
                                </div>

                                </div>

                                <div id="navigation">

                                <!--Labels RAD1: -->

                                <asp:Label CssClass="label1 move" ID="label1_Artist"
runat="server" Text="Artist"></asp:Label>

                                <asp:Label CssClass="label1 move" ID="label2_Album"
runat="server" Text="Titel"></asp:Label>

                                <asp:Label CssClass="label1 move" ID="label3_Format"
runat="server" Text="Format"></asp:Label>

                                <asp:Label CssClass="label1 move" ID="label4_Press"
runat="server" Text="Press"></asp:Label>

                                <asp:Label CssClass="label1 move" ID="label5_Kommentar"
runat="server" Text="Kommentar"></asp:Label>

                                <asp:Label CssClass="label1 move" ID="label6_ar"
runat="server" Text="År"></asp:Label>

                                <!--Texttrutor RAD1: -->

                                <asp:TextBox CssClass="move text1" ID="text3_Artist"
MaxLength="50"
                                runat="server" Size="15"></asp:TextBox>

```

```

runat="server"
Enabled="true"
        <asp:RequiredFieldValidator ID="vldArtistNotEmpty"
            Display="None" ControlToValidate="text3_Artist"
            EnableClientScript="false" ValidationGroup="x"/>

        <asp:TextBox CssClass="move text1" ID="text4_Album"
            MaxLength="80" runat="server" Size="15"></asp:TextBox>

        <asp:DropDownList CssClass="move text1" ID="text5_Format"
            runat="server" Size="1">
            <asp:ListItem />
            <asp:ListItem Value="CD" />
            <asp:ListItem Value="CDM" />
            <asp:ListItem Value="CDS" />
            <asp:ListItem Value="2CD" />
            <asp:ListItem Value="LCDM" />
            <asp:ListItem Value="XLCDM" />
            <asp:ListItem Value="DVDS" />
            <asp:ListItem Value="CD+DVD" />
            <asp:ListItem Value="DVD/SACD" />
            <asp:ListItem Value="CDM/DVD" />
            <asp:ListItem Value="dEP" />
            <asp:ListItem Value="LP" />
            <asp:ListItem Value="2LP" />
            <asp:ListItem Value="BOX" />
            <asp:ListItem Value="12" />
            <asp:ListItem Value="L12" />
            <asp:ListItem Value="XL12" />
            <asp:ListItem Value="L12num" />
            <asp:ListItem Value="TwinSetL12" />
            <asp:ListItem Value="TwinSetL12num" />
            <asp:ListItem Value="Promo" />
            <asp:ListItem Value="7" />
            <asp:ListItem Value="aEP" />
            <asp:ListItem Value="mp3" />
        </asp:DropDownList>

        <asp:DropDownList CssClass="move text1" ID="text6_Press"
            runat="server" Size="1">
            <asp:ListItem />
            <asp:ListItem Value="at" />
            <asp:ListItem Value="be" />
            <asp:ListItem Value="ca" />
            <asp:ListItem Value="de" />
            <asp:ListItem Value="dk" />
            <asp:ListItem Value="eu" />
            <asp:ListItem Value="fr" />
            <asp:ListItem Value="holl" />
            <asp:ListItem Value="it" />
            <asp:ListItem Value="se" />
            <asp:ListItem Value="uk" />
            <asp:ListItem Value="ukorg" />
            <asp:ListItem Value="us" />
        </asp:DropDownList>

```

```

        <asp:DropDownList CssClass="move text1" ID="text8_ar"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="1965" />
        <asp:ListItem Value="1966" />
        <asp:ListItem Value="1967" />
        <asp:ListItem Value="1968" />
        <asp:ListItem Value="1969" />
        <asp:ListItem Value="1970" />
        <asp:ListItem Value="1971" />
        <asp:ListItem Value="1972" />
        <asp:ListItem Value="1973" />
        <asp:ListItem Value="1974" />
        <asp:ListItem Value="1975" />
        <asp:ListItem Value="1976" />
        <asp:ListItem Value="1977" />
        <asp:ListItem Value="1978" />
        <asp:ListItem Value="1979" />
        <asp:ListItem Value="1980" />
        <asp:ListItem Value="1981" />
        <asp:ListItem Value="1982" />
        <asp:ListItem Value="1983" />
        <asp:ListItem Value="1984" />
        <asp:ListItem Value="1985" />
        <asp:ListItem Value="1986" />
        <asp:ListItem Value="1987" />
        <asp:ListItem Value="1988" />
        <asp:ListItem Value="1989" />
        <asp:ListItem Value="1990" />
        <asp:ListItem Value="1991" />
        <asp:ListItem Value="1992" />
        <asp:ListItem Value="1993" />
        <asp:ListItem Value="1994" />
        <asp:ListItem Value="1995" />
        <asp:ListItem Value="1996" />
        <asp:ListItem Value="1997" />
        <asp:ListItem Value="1998" />
        <asp:ListItem Value="1999" />
        <asp:ListItem Value="2000" />
        <asp:ListItem Value="2001" />
        <asp:ListItem Value="2002" />
        <asp:ListItem Value="2003" />
        <asp:ListItem Value="2004" />
        <asp:ListItem Value="2005" />
        <asp:ListItem Value="2006" />
        <asp:ListItem Value="2007" />
        <asp:ListItem Value="2008" />
        <asp:ListItem Value="2009" />
        <asp:ListItem Value="2010" />
        <asp:ListItem Value="2011" />
        <asp:ListItem Value="2012" />
        <asp:ListItem Value="2013" />
        <asp:ListItem Value="2014" />
        <asp:ListItem Value="2015" />
        <asp:ListItem Value="2016" />
        <asp:ListItem Value="2017" />
        <asp:ListItem Value="2018" />
        <asp:ListItem Value="2019" />
        <asp:ListItem Value="2020" />

</asp:DropDownList>

```

```

        <asp:TextBox CssClass="move text1" ID="text7_Kommentar"
MaxLength="1000" runat="server" Size="15"></asp:TextBox>

<!--Labels RAD2: -->

        <asp:Label CssClass="label2 move" ID="label7_VPfr"
runat="server" Text="VPfr"></asp:Label>

        <asp:Label CssClass="label2 move" ID="label8_VPtill"
runat="server" Text="VPtill"></asp:Label>

        <asp:Label CssClass="label2 move" ID="label9_Ursland"
runat="server" Text="Ursland"></asp:Label>

        <asp:Label CssClass="label2 move" ID="label10_Ursstad"
runat="server" Text="Ursstad"></asp:Label>

        <asp:Label CssClass="label2 move" ID="label11_Musiktyp"
runat="server" Text="Musiktyp"></asp:Label>

        <asp:Label CssClass="label2 move" ID="label12_Kortbio"
runat="server" Text="Kortbio"></asp:Label>

        <asp:Label CssClass="label2 move" ID="label13_Inm_dat"
runat="server" Text="Inm_dat"></asp:Label>

        <asp:Label CssClass="label2 move" ID="label14_Kop_grad"
runat="server" Text="Kgd"></asp:Label>

        <asp:Label CssClass="label2 move" ID="label15_Kop_kat"
runat="server" Text="Kkat"></asp:Label>

        <asp:Label CssClass="label2 move" ID="label16_Ca_pris"
runat="server" Text="CPris"></asp:Label>

<!--Texttrutor RAD2: -->

        <asp:DropDownList CssClass="move text2" ID="text9_VPfr"
runat="server" Size="1">
            <asp:ListItem />
            <asp:ListItem Value="1965" />
            <asp:ListItem Value="1966" />
            <asp:ListItem Value="1967" />
            <asp:ListItem Value="1968" />
            <asp:ListItem Value="1969" />
            <asp:ListItem Value="1970" />
            <asp:ListItem Value="1971" />
            <asp:ListItem Value="1972" />
            <asp:ListItem Value="1973" />
            <asp:ListItem Value="1974" />
            <asp:ListItem Value="1975" />
            <asp:ListItem Value="1976" />
            <asp:ListItem Value="1977" />
            <asp:ListItem Value="1978" />
            <asp:ListItem Value="1979" />
            <asp:ListItem Value="1980" />
            <asp:ListItem Value="1981" />

```

```

        <asp:ListItem Value="1982" />
        <asp:ListItem Value="1983" />
        <asp:ListItem Value="1984" />
        <asp:ListItem Value="1985" />
        <asp:ListItem Value="1986" />
        <asp:ListItem Value="1987" />
        <asp:ListItem Value="1988" />
        <asp:ListItem Value="1989" />
        <asp:ListItem Value="1990" />
        <asp:ListItem Value="1991" />
        <asp:ListItem Value="1992" />
        <asp:ListItem Value="1993" />
        <asp:ListItem Value="1994" />
        <asp:ListItem Value="1995" />
        <asp:ListItem Value="1996" />
        <asp:ListItem Value="1997" />
        <asp:ListItem Value="1998" />
        <asp:ListItem Value="1999" />
        <asp:ListItem Value="2000" />
        <asp:ListItem Value="2001" />
        <asp:ListItem Value="2002" />
        <asp:ListItem Value="2003" />
        <asp:ListItem Value="2004" />
        <asp:ListItem Value="2005" />
        <asp:ListItem Value="2006" />
        <asp:ListItem Value="2007" />
        <asp:ListItem Value="2008" />
        <asp:ListItem Value="2009" />
        <asp:ListItem Value="2010" />
        <asp:ListItem Value="2011" />
        <asp:ListItem Value="2012" />
        <asp:ListItem Value="2013" />
        <asp:ListItem Value="2014" />
        <asp:ListItem Value="2015" />
        <asp:ListItem Value="2016" />
        <asp:ListItem Value="2017" />
        <asp:ListItem Value="2018" />
        <asp:ListItem Value="2019" />
        <asp:ListItem Value="2020" />
    </asp:DropDownList>

    <asp:DropDownList CssClass="move text2" ID="text10_VPtill"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="active" />
        <asp:ListItem Value="1965" />
        <asp:ListItem Value="1966" />
        <asp:ListItem Value="1967" />
        <asp:ListItem Value="1968" />
        <asp:ListItem Value="1969" />
        <asp:ListItem Value="1970" />
        <asp:ListItem Value="1971" />
        <asp:ListItem Value="1972" />
        <asp:ListItem Value="1973" />
        <asp:ListItem Value="1974" />
        <asp:ListItem Value="1975" />
        <asp:ListItem Value="1976" />
        <asp:ListItem Value="1977" />
        <asp:ListItem Value="1978" />
        <asp:ListItem Value="1979" />
        <asp:ListItem Value="1980" />

```

```

        <asp:ListItem Value="1981" />
        <asp:ListItem Value="1982" />
        <asp:ListItem Value="1983" />
        <asp:ListItem Value="1984" />
        <asp:ListItem Value="1985" />
        <asp:ListItem Value="1986" />
        <asp:ListItem Value="1987" />
        <asp:ListItem Value="1988" />
        <asp:ListItem Value="1989" />
        <asp:ListItem Value="1990" />
        <asp:ListItem Value="1991" />
        <asp:ListItem Value="1992" />
        <asp:ListItem Value="1993" />
        <asp:ListItem Value="1994" />
        <asp:ListItem Value="1995" />
        <asp:ListItem Value="1996" />
        <asp:ListItem Value="1997" />
        <asp:ListItem Value="1998" />
        <asp:ListItem Value="1999" />
        <asp:ListItem Value="2000" />
        <asp:ListItem Value="2001" />
        <asp:ListItem Value="2002" />
        <asp:ListItem Value="2003" />
        <asp:ListItem Value="2004" />
        <asp:ListItem Value="2005" />
        <asp:ListItem Value="2006" />
        <asp:ListItem Value="2007" />
        <asp:ListItem Value="2008" />
        <asp:ListItem Value="2009" />
        <asp:ListItem Value="2010" />
        <asp:ListItem Value="2011" />
        <asp:ListItem Value="2012" />
        <asp:ListItem Value="2013" />
        <asp:ListItem Value="2014" />
        <asp:ListItem Value="2015" />
        <asp:ListItem Value="2016" />
        <asp:ListItem Value="2017" />
        <asp:ListItem Value="2018" />
        <asp:ListItem Value="2019" />
        <asp:ListItem Value="2020" />
    </asp:DropDownList>

    <asp:DropDownList CssClass="move text2" ID="text11_Ursland"
runat="server" Size="1">
    <asp:ListItem />
    <asp:ListItem Value="at" />
        <asp:ListItem Value="be" />
        <asp:ListItem Value="ca" />
        <asp:ListItem Value="de" />
        <asp:ListItem Value="dk" />
        <asp:ListItem Value="eu" />
        <asp:ListItem Value="fr" />
        <asp:ListItem Value="holl" />
        <asp:ListItem Value="it" />
        <asp:ListItem Value="se" />
        <asp:ListItem Value="uk" />
        <asp:ListItem Value="ukorg" />
        <asp:ListItem Value="us" />
    </asp:DropDownList>

```



```

        <asp:TextBox CssClass="move text2" ID="text12_Ursstad"
MaxLength="50" runat="server" Size="6"></asp:TextBox>

        <asp:RegularExpressionValidator
id="vldArtistOnlyLetters" runat="server"
        Display="None" ValidationExpression="^[a-öA-Ö\s.]*$"
        ControlToValidate="text12_Ursstad" Enabled="true"
        EnableClientScript="false" ValidationGroup="x"/>

        <asp:DropDownList CssClass="move text2"
ID="text13_Musiktyp" runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="Cyberpunk" />
        <asp:ListItem Value="EBM" />
        <asp:ListItem Value="EuroDisco" />
        <asp:ListItem Value="Gothrock" />
        <asp:ListItem Value="HipHop" />
        <asp:ListItem Value="Industri" />
        <asp:ListItem Value="New Wave" />
        <asp:ListItem Value="Pop" />
        <asp:ListItem Value="Punk" />
        <asp:ListItem Value="RnB" />
        <asp:ListItem Value="Rock" />
        <asp:ListItem Value="Synth" />
        <asp:ListItem Value="Synthpop" />
        </asp:DropDownList>

        <asp:TextBox CssClass="move text2" ID="text14_Kortbio"
MaxLength="6000" runat="server" Size="6"></asp:TextBox>

        <asp:TextBox CssClass="move text2" ID="text15_Inm_dat"
MaxLength="10"
                runat="server" Size="6"></asp:TextBox>

        <asp:DropDownList CssClass="move text2"
ID="text16_Kop_grad" runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="1" />
        <asp:ListItem Value="2" />
        <asp:ListItem Value="3" />
        <asp:ListItem Value="4" />
        <asp:ListItem Value="5" />
        </asp:DropDownList>

        <asp:DropDownList CssClass="move text2" ID="text17_Kop_kat"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="Ny" />
        <asp:ListItem Value="Beg" />
        <asp:ListItem Value="Rea" />
        </asp:DropDownList>

        <asp:TextBox CssClass="move text2" ID="text18_Ca_pris"
MaxLength="4"
                runat="server" Size="1"
                ></asp:TextBox>

        <!--Knappar RAD1: -->

```

```

<asp:Button ID="button_visa_databas" runat="server"
  CssClass="input move rad1" Text="Visa_TB"
  onclick="button_visa_databas_Click"
  CausesValidation="False" />

<asp:Button ID="button_lagg_till" runat="server"
  CssClass="input move rad1" Text="Lägg_till"
  onclick="button_lagg_till_Click"
  CausesValidation="true" ValidationGroup="x"/>

<asp:Button ID="button_ta_bort" runat="server"
  CssClass="input move rad1" Text="Ta_bort"
  onclick="button_ta_bort_Click"
  CausesValidation="False" />

<asp:Button ID="button_uppdatera" runat="server"
  CssClass="input move rad1" Text="Uppdatera"
  onclick="button_uppdatera_Click"
  CausesValidation="False" />

<asp:Button ID="button_sok" runat="server"
  CssClass="input move rad1" Text="Sök"
  onclick="button_sok_Click"
  CausesValidation="False" />

<!--Knappar - RAD2: -->

<asp:Button ID="btn21" runat="server"
  CssClass="input move rad2" Text="Hämta"
  onclick="button_fetch_Click"
  CausesValidation="False"/>

<asp:Button ID="btn22" runat="server"
  CssClass="input move rad2" Text="Kortbio"
  onclick="button_kortbiopanel_Click"
  CausesValidation="False"/>

<asp:Button ID="btn23" runat="server"
  CssClass="input move rad2" Text="Börja_om"
  OnClientClick="this.form.reset();return false;"
  onclick="button_Borja_om_Click"
  CausesValidation="False" />

<asp:Button ID="btn24" runat="server"
  CssClass="input move rad2" Text="Omstart_TB"
  onclick="button_Refresh_TB_Click"
  CausesValidation="False" />

<asp:Button ID="btn25" runat="server"
  CssClass="input move rad2" Text="HOLDER"
  CausesValidation="False"/>

```

```

        <!-- Tabellvals-radioknappar inkl. labels (i form av div): -->

        <div class="label3 move" id="label13_TBskivor">Skivtabell</div>

        <asp:RadioButton ID="optSkivtabell" runat="server"
GroupName="radioval" TextAlign="Right"
AutoPostBack="true" CssClass="optSkivtabell" Checked="false"
OnCheckedChanged="Skivtabell_CheckedChanged" />

        <div class="label3 move"
id="label14_TBartist">Artisttabell</div>
        <asp:RadioButton ID="optArtisttabell" runat="server"
GroupName="radioval" TextAlign="Right"
AutoPostBack="true" CssClass="optArtisttabell"
OnCheckedChanged="Artisttabell_CheckedChanged"
Checked="true" />

        <div class="label3 move" id="label15_TBinkop">Inköpstabell</div>
        <asp:RadioButton ID="optInköpstabell" runat="server"
GroupName="radioval" TextAlign="Right"
AutoPostBack="true" CssClass="optInköpstabell"
OnCheckedChanged="Inköpstabell_CheckedChanged"
Checked="false" />

    </div>

</div>
</form>
</body>
</html>

```

Bakomliggande kod för artisttabellen där anrop till olika komponenter med olika funktioner sker. Mesta av felhanteringen sköts här och felmeddelanden ges till användaren när sådana har uppkommit.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Collections;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

using DB_Callcode_Artist.Artist;

public partial class Default2 : System.Web.UI.Page
{
    //Instans för rensning av display och gridviews
    Clear_Artist clr = new Clear_Artist();

    protected void Page_Load(object sender, EventArgs e)
    {}

    //Knappar Rad 1:

    //Hämtar hela databasen:
    public void button_visa_databas_Click(object sender, EventArgs e)
    {
        //Leta upp kontrollen:
        UserControl gridArtister_Artist_UCtrl =
        (UserControl)Page.FindControl("gridArtister_Artist_UCtrl");

        GridView gridArtister_Artist =
        (GridView)gridArtister_Artist_UCtrl.FindControl("gridArtister_Artist");

        Visa_databas vdb = new Visa_databas();
        vdb.button_visa_databas(gridArtister_Artist, this.Page);
    }

    //Lägger till data i databas:
    protected void button_lagg_till_Click(object sender, EventArgs e)
    {
        try
        {
```

```

        Lagg_till lt = new Lagg_till();
        lt.button_lagg_till(display, text3_Artist, text9_VPfr,
text10_VPtill,
        text11_Ursland, text12_Ursstad, text13_Musiktyp, text14_Kortbio,
        vldArtistNotEmpty, vldArtistOnlyLetters, this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess = "<h2>[Page]Artist.Lagg_till:
FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }

    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess = "<h2>[Page]Artist.Lagg_till:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }

    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[Page]Artist.Lagg_till:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Tar bort data från databas:
protected void button_ta_bort_Click(object sender, EventArgs e)
{
    //Leta upp kontroller:
    GridView grdResult_Artist =
(Gridview)grdResult_Artist_UCtrl.FindControl("grdResult_Artist");
    GridView gridArtister_Artist =
(Gridview)gridArtister_Artist_UCtrl.FindControl("gridArtister_Artist");

    try
    {
        Ta_bort tb = new Ta_bort();
        tb.button_ta_bort(gridArtister_Artist, grdResult_Artist,
display, this.Page);
    }
    catch (FormatException err)

```

```

{
    //Rensar display från text och gridviews
    clr.Clean_surfaces_Artist(this.Page);

    string mess = "<h2>[Page]Artist.Ta_bort:FormatException</h2>";
    mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
    display.InnerHtml = mess;
}
catch (MySqlException err)
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces_Artist(this.Page);

    string mess = "<h2>[Page]Artist.Ta_bort:MySqlException</h2>";
    mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
    display.InnerHtml = mess;
}
catch (System.Exception err)
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces_Artist(this.Page);

    string mess = "<h2>[Page]Artist.Ta_bort:System.Exception</h2>";
    mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
    display.InnerHtml = mess;
}
}

//Uppdaterar databas:
protected void button_uppdatera_Click(object sender, EventArgs e)
{
    //Leta upp kontrollern:
    GridView grdResult_Artist =
(Gridview)grdResult_Artist_UCtrl.FindControl("grdResult_Artist");
    GridView gridArtister_Artist =
(Gridview)gridArtister_Artist_UCtrl.FindControl("gridArtister_Artist");

    try
    {
        Uppdatera upd = new Uppdatera();
        upd.button_uppdatera(gridArtister_Artist, grdResult_Artist,
display, this.Page, text3_Artist, text9_VPfr,
text10_VPtill, text11_Ursland, text12_Ursstad, text13_Musiktyp,
text14_Kortbio);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[Page]Artist.Uppdatera:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {

```

```

        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess = "<h2>[Page]Artist.Uppdatera:MySQLException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[Page]Artist.Uppdatera:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

```

//Söker i databas:
protected void button_sok_Click(object sender, EventArgs e)
{
    //Leta upp kontrollern:
    GridView grdResult_Artist =
(GridView)grdResult_Artist_UCtrl.FindControl("grdResult_Artist");

    //Skapar instans av objekt för delayvärde och sök-anrop:
    Sok sk = new Sok();

    //Anger antal sekunder som sökningen skall vara i cache-objektet:
    sk.Delay = 20;

    try
    {
        sk.button_sok(grdResult_Artist, this.Page, display,
            text3_Artist, text9_VPfr, text10_VPtill, text11_Ursland,
            text12_Ursstad, text13_Musiktyp, text14_Kortbio);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[Page]Artist.Sok_Artist:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySQLException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);
    }
}

```

```

        string mess =
"<h2>[Page]Artist.Sok_Artist:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess = "<h2>[Page]Artist.Sok:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

//Knappar Rad 2:

```

//Hämtar en rad från aktuell tabell som skall uppdateras:
protected void button_fetch_Click(object sender, EventArgs e)
{
    //Leta upp kontroller:
    GridView grdResult_Artist =
(Gridview)grdResult_Artist_UCtrl.FindControl("grdResult_Artist");
    GridView gridArtister_Artist =
(Gridview)gridArtister_Artist_UCtrl.FindControl("gridArtister_Artist");

    try
    {
        Fetch fch = new Fetch();
        fch.button_fetch(gridArtister_Artist, grdResult_Artist,
            display, display2, this.Page, text3_Artist, text9_VPfr,
            text10_VPtill,
            text11_Ursland, text12_Ursstad, text13_Musiktyp,
            text14_Kortbio);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess = "<h2>[Page]Artist.Fetch:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
    }
}

```



```

        clr.Clean_surfaces_Artist(this.Page);

        string mess = "<h2>[Page]Artist.Fetch:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess = "<h2>[Page]Artist.Fetch:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Visar panel för inmatning av en artist kort-biografi:
protected void button_kortbiopanel_Click(object sender, EventArgs e)
{
    //Leta upp kontrollen:
    Panel pnlKortbio =
    (Panel)pnlKortbio_UCtrl.FindControl("pnlKortbio");

    try
    {
        Visa_KortBioPanel kbpnl = new Visa_KortBioPanel();
        kbpnl.button_kortbiopanel(pnlKortbio, this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
        "<h2>[Page]Artist.kortbiopanel:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
        "<h2>[Page]Artist.kortbiopanel:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);
    }
}

```

```

        string mess =
"<h2>[Page]Artist.kortbiopanel:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Raderar innehållet i alla textrutor och dropdownmenyer:
protected void button_Borja_om_Click(object sender, EventArgs e)
{
    Borja_om bro = new Borja_om();
    bro.button_Borja_om();
}

//Fungerar som en omstart och rensar alla inmatningsrutor odyl +
display:
protected void button_Refresh_TB_Click(object sender, EventArgs e)
{
    HttpServerUtility server = this.Server;
    String retursida = "~/Default2.aspx";

    try
    {
        Refresh_TB rtb = new Refresh_TB();
        rtb.button_Refresh_TB(server, retursida);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[Page]Artist.Refresh_TB:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[Page]Artist.Refresh_TB:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[Page]Artist.Refresh_TBSystem.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
    }
}

```

```

        display.InnerHtml = mess;
    }
}

//Växla mellan tabeller:

protected void Skivtabell_CheckedChanged(object sender, EventArgs e)
{
    IDictionaryEnumerator CacheEnum =
HttpRuntime.Cache.GetEnumerator();
    while (CacheEnum.MoveNext())
    {
        string key = CacheEnum.Key.ToString();
        HttpRuntime.Cache.Remove(key);
    }
    Server.Transfer("~/Default.aspx");
}

protected void Artisttabell_CheckedChanged(object sender, EventArgs e)
{
    IDictionaryEnumerator CacheEnum =
HttpRuntime.Cache.GetEnumerator();
    while (CacheEnum.MoveNext())
    {
        string key = CacheEnum.Key.ToString();
        HttpRuntime.Cache.Remove(key);
    }
    Server.Transfer("~/Default2.aspx");
}

protected void Inkoptabell_CheckedChanged(object sender, EventArgs e)
{
    IDictionaryEnumerator CacheEnum =
HttpRuntime.Cache.GetEnumerator();
    while (CacheEnum.MoveNext())
    {
        string key = CacheEnum.Key.ToString();
        HttpRuntime.Cache.Remove(key);
    }
    Server.Transfer("~/Default3.aspx");
}
}

```

Bildar strukturen för Inkopstabellen.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default3.aspx.cs"
Inherits="Default3" ValidateRequest="false" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>SkivorNETAdmin</title>

    <!-- JavaScript: -->
    <script type="text/javascript" src="javascript/browser.js"></script>
    <script type="text/javascript" src="javascript/datum.js"></script>

</head>
<body onload="datum(); setInterval('datum()', 1000 )">
    <form id="form1" runat="server">

        <div id="container">

            <div id="header">

                <!--Datumruta; Visning av Datum och tid: -->
                <div id="datumruta">

                    <div id="clock">&nbsp;</div>

                </div>

            </div>

            <div id="content">

                <!-- Display-ram börjar: -->
                <div class="ruta">

                    <!--Panel att "skriva" till: -->
                    <asp:Panel ID="frame" runat="server">

                        <!-- För att visa information/data som skall stå över
kontroll: -->
                        <span id="display" runat="server"
class="mess"></span>

                        <!-- GridView för att visa artistdata (i
usercontrol): -->
                        <TF:UGridResult_Inkop ID="grdResult_Inkop_UCtrl"
Visible="true" runat="server" />

                    </asp:Panel>

                </div>

            </div>

        </div>

    </form>

</body>
</html>
```

```

        <!-- GridView för att visa resultat från Hämta-
funktion och sökning (i usercontrol): -->
        <TF:UGridArtistdata_Inkop
ID="grd_Artistdata_Inkop_UCtrl" Visible="true" runat="server" />

-->
        <!-- GridView som visar hel tabell inkl paging etc:
-->
        <TF:UGridArtister_Inkop
ID="UGridArtister_Inkop_UCtrl" Visible="true" runat="server" />

        <!-- Panel för administrering av omfattande
kommentarer: -->
        <TF:UPnlKommentar_Inkop
ID="pnlKommentar_Inkop_UCtrl" Visible="true" runat="server" />

        <!-- För att visa information/data som skall stå under
kontroll: -->
        <span id="display2" runat="server" class="mess"></span>

        </asp:Panel>

        <!--Slut på display-ram: -->
    </div>

</div>

<div id="navigation">

        <!--Labels RAD1: -->

        <asp:Label CssClass="label1 move" ID="label1_Artist"
runat="server" Text="Artist"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label2_Album"
runat="server" Text="Titel"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label3_Format"
runat="server" Text="Format"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label4_Press"
runat="server" Text="Press"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label5_Kommentar"
runat="server" Text="Kommentar"></asp:Label>

        <asp:Label CssClass="label1 move" ID="label6_ar"
runat="server" Text="År"></asp:Label>

        <!--Texttrutor RAD1: -->

```

```

        <asp:TextBox CssClass="move text1" ID="text3_Artist"
MaxLength="50"
        runat="server" Size="15"></asp:TextBox>

        <asp:RegularExpressionValidator id="vldArtistOnlyNum"
runat="server"
        Display="None" ValidationExpression="^([0-9]{0,4})$"
        ControlToValidate="text3_Artist" Enabled="true"
        EnableClientScript="false" ValidationGroup="x"/>

        <asp:RequiredFieldValidator ID="vldArtistNotEmpty"
runat="server"
        Display="None" ControlToValidate="text3_Artist"
Enabled="true"
        EnableClientScript="false" ValidationGroup="x"/>

        <asp:RangeValidator ID="vldArtistValidRange"
runat="server"
        Display="None" ControlToValidate="text3_Artist"
        Enabled="true" EnableClientScript="false"
Type="Integer"
        ValidationGroup="x"/>

        <asp:TextBox CssClass="move text1" ID="text4_Album"
MaxLength="80" runat="server" Size="15"></asp:TextBox>

        <asp:DropDownList CssClass="move text1" ID="text5_Format"
runat="server" Size="1">
            <asp:ListItem />
            <asp:ListItem Value="CD" />
            <asp:ListItem Value="CDM" />
            <asp:ListItem Value="CDS" />
            <asp:ListItem Value="2CD" />
            <asp:ListItem Value="LCDM" />
            <asp:ListItem Value="XLCDM" />
            <asp:ListItem Value="DVDS" />
            <asp:ListItem Value="CD+DVD" />
            <asp:ListItem Value="DVD/SACD" />
            <asp:ListItem Value="CDM/DVD" />
            <asp:ListItem Value="dEP" />
            <asp:ListItem Value="LP" />
            <asp:ListItem Value="2LP" />
            <asp:ListItem Value="BOX" />
            <asp:ListItem Value="12" />
            <asp:ListItem Value="L12" />
            <asp:ListItem Value="XL12" />
            <asp:ListItem Value="L12num" />
            <asp:ListItem Value="TwinSetL12" />
            <asp:ListItem Value="TwinSetL12num" />
            <asp:ListItem Value="Promo" />
            <asp:ListItem Value="7" />
            <asp:ListItem Value="aEP" />
            <asp:ListItem Value="mp3" />
        </asp:DropDownList>

        <asp:DropDownList CssClass="move text1" ID="text6_Press"
runat="server" Size="1">
            <asp:ListItem />
            <asp:ListItem Value="at" />

```

```

        <asp:ListItem Value="be" />
        <asp:ListItem Value="ca" />
        <asp:ListItem Value="de" />
        <asp:ListItem Value="dk" />
        <asp:ListItem Value="eu" />
        <asp:ListItem Value="fr" />
        <asp:ListItem Value="holl" />
        <asp:ListItem Value="it" />
        <asp:ListItem Value="se" />
        <asp:ListItem Value="uk" />
        <asp:ListItem Value="ukorg" />
        <asp:ListItem Value="us" />
    </asp:DropDownList>

    <asp:DropDownList CssClass="move text1" ID="text8_ar"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="1965" />
        <asp:ListItem Value="1966" />
        <asp:ListItem Value="1967" />
        <asp:ListItem Value="1968" />
        <asp:ListItem Value="1969" />
        <asp:ListItem Value="1970" />
        <asp:ListItem Value="1971" />
        <asp:ListItem Value="1972" />
        <asp:ListItem Value="1973" />
        <asp:ListItem Value="1974" />
        <asp:ListItem Value="1975" />
        <asp:ListItem Value="1976" />
        <asp:ListItem Value="1977" />
        <asp:ListItem Value="1978" />
        <asp:ListItem Value="1979" />
        <asp:ListItem Value="1980" />
        <asp:ListItem Value="1981" />
        <asp:ListItem Value="1982" />
        <asp:ListItem Value="1983" />
        <asp:ListItem Value="1984" />
        <asp:ListItem Value="1985" />
        <asp:ListItem Value="1986" />
        <asp:ListItem Value="1987" />
        <asp:ListItem Value="1988" />
        <asp:ListItem Value="1989" />
        <asp:ListItem Value="1990" />
        <asp:ListItem Value="1991" />
        <asp:ListItem Value="1992" />
        <asp:ListItem Value="1993" />
        <asp:ListItem Value="1994" />
        <asp:ListItem Value="1995" />
        <asp:ListItem Value="1996" />
        <asp:ListItem Value="1997" />
        <asp:ListItem Value="1998" />
        <asp:ListItem Value="1999" />
        <asp:ListItem Value="2000" />
        <asp:ListItem Value="2001" />
        <asp:ListItem Value="2002" />
        <asp:ListItem Value="2003" />
        <asp:ListItem Value="2004" />
        <asp:ListItem Value="2005" />
        <asp:ListItem Value="2006" />
        <asp:ListItem Value="2007" />

```

```

        <asp:ListItem Value="2008" />
        <asp:ListItem Value="2009" />
        <asp:ListItem Value="2010" />
        <asp:ListItem Value="2011" />
        <asp:ListItem Value="2012" />
        <asp:ListItem Value="2013" />
        <asp:ListItem Value="2014" />
        <asp:ListItem Value="2015" />
        <asp:ListItem Value="2016" />
        <asp:ListItem Value="2017" />
        <asp:ListItem Value="2018" />
        <asp:ListItem Value="2019" />
        <asp:ListItem Value="2020" />

    </asp:DropDownList>

    <asp:TextBox CssClass="move text1" ID="text7_Kommentar"
MaxLength="1000" runat="server" Size="15"></asp:TextBox>

<!--Labels RAD2: -->

    <asp:Label CssClass="label2 move" ID="label7_VPfr"
runat="server" Text="VPfr"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label8_VPtill"
runat="server" Text="VPtill"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label9_Ursland"
runat="server" Text="Ursland"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label10_Ursstad"
runat="server" Text="Ursstad"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label11_Musiktyp"
runat="server" Text="Musiktyp"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label12_Kortbio"
runat="server" Text="Kortbio"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label13_Inm_dat"
runat="server" Text="Inm_dat"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label14_Kop_grad"
runat="server" Text="Kgd"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label15_Kop_kat"
runat="server" Text="Kkat"></asp:Label>

    <asp:Label CssClass="label2 move" ID="label16_Ca_pris"
runat="server" Text="CPris"></asp:Label>

<!--Texttrutor RAD2: -->

    <asp:DropDownList CssClass="move text2" ID="text9_VPfr"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="1965" />
        <asp:ListItem Value="1966" />

```



```

        <asp:ListItem Value="1967" />
        <asp:ListItem Value="1968" />
        <asp:ListItem Value="1969" />
        <asp:ListItem Value="1970" />
        <asp:ListItem Value="1971" />
        <asp:ListItem Value="1972" />
        <asp:ListItem Value="1973" />
        <asp:ListItem Value="1974" />
        <asp:ListItem Value="1975" />
        <asp:ListItem Value="1976" />
        <asp:ListItem Value="1977" />
        <asp:ListItem Value="1978" />
        <asp:ListItem Value="1979" />
        <asp:ListItem Value="1980" />
        <asp:ListItem Value="1981" />
        <asp:ListItem Value="1982" />
        <asp:ListItem Value="1983" />
        <asp:ListItem Value="1984" />
        <asp:ListItem Value="1985" />
        <asp:ListItem Value="1986" />
        <asp:ListItem Value="1987" />
        <asp:ListItem Value="1988" />
        <asp:ListItem Value="1989" />
        <asp:ListItem Value="1990" />
        <asp:ListItem Value="1991" />
        <asp:ListItem Value="1992" />
        <asp:ListItem Value="1993" />
        <asp:ListItem Value="1994" />
        <asp:ListItem Value="1995" />
        <asp:ListItem Value="1996" />
        <asp:ListItem Value="1997" />
        <asp:ListItem Value="1998" />
        <asp:ListItem Value="1999" />
        <asp:ListItem Value="2000" />
        <asp:ListItem Value="2001" />
        <asp:ListItem Value="2002" />
        <asp:ListItem Value="2003" />
        <asp:ListItem Value="2004" />
        <asp:ListItem Value="2005" />
        <asp:ListItem Value="2006" />
        <asp:ListItem Value="2007" />
        <asp:ListItem Value="2008" />
        <asp:ListItem Value="2009" />
        <asp:ListItem Value="2010" />
        <asp:ListItem Value="2011" />
        <asp:ListItem Value="2012" />
        <asp:ListItem Value="2013" />
        <asp:ListItem Value="2014" />
        <asp:ListItem Value="2015" />
        <asp:ListItem Value="2016" />
        <asp:ListItem Value="2017" />
        <asp:ListItem Value="2018" />
        <asp:ListItem Value="2019" />
        <asp:ListItem Value="2020" />
    </asp:DropDownList>

    <asp:DropDownList CssClass="move text2" ID="text10_VPtill"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="active" />
        <asp:ListItem Value="1965" />

```

```

        <asp:ListItem Value="1966" />
        <asp:ListItem Value="1967" />
        <asp:ListItem Value="1968" />
        <asp:ListItem Value="1969" />
        <asp:ListItem Value="1970" />
        <asp:ListItem Value="1971" />
        <asp:ListItem Value="1972" />
        <asp:ListItem Value="1973" />
        <asp:ListItem Value="1974" />
        <asp:ListItem Value="1975" />
        <asp:ListItem Value="1976" />
        <asp:ListItem Value="1977" />
        <asp:ListItem Value="1978" />
        <asp:ListItem Value="1979" />
        <asp:ListItem Value="1980" />
        <asp:ListItem Value="1981" />
        <asp:ListItem Value="1982" />
        <asp:ListItem Value="1983" />
        <asp:ListItem Value="1984" />
        <asp:ListItem Value="1985" />
        <asp:ListItem Value="1986" />
        <asp:ListItem Value="1987" />
        <asp:ListItem Value="1988" />
        <asp:ListItem Value="1989" />
        <asp:ListItem Value="1990" />
        <asp:ListItem Value="1991" />
        <asp:ListItem Value="1992" />
        <asp:ListItem Value="1993" />
        <asp:ListItem Value="1994" />
        <asp:ListItem Value="1995" />
        <asp:ListItem Value="1996" />
        <asp:ListItem Value="1997" />
        <asp:ListItem Value="1998" />
        <asp:ListItem Value="1999" />
        <asp:ListItem Value="2000" />
        <asp:ListItem Value="2001" />
        <asp:ListItem Value="2002" />
        <asp:ListItem Value="2003" />
        <asp:ListItem Value="2004" />
        <asp:ListItem Value="2005" />
        <asp:ListItem Value="2006" />
        <asp:ListItem Value="2007" />
        <asp:ListItem Value="2008" />
        <asp:ListItem Value="2009" />
        <asp:ListItem Value="2010" />
        <asp:ListItem Value="2011" />
        <asp:ListItem Value="2012" />
        <asp:ListItem Value="2013" />
        <asp:ListItem Value="2014" />
        <asp:ListItem Value="2015" />
        <asp:ListItem Value="2016" />
        <asp:ListItem Value="2017" />
        <asp:ListItem Value="2018" />
        <asp:ListItem Value="2019" />
        <asp:ListItem Value="2020" />
    </asp:DropDownList>

    <asp:DropDownList CssClass="move text2" ID="text11_Ursland"
runat="server" Size="1">
        <asp:ListItem />
        <asp:ListItem Value="at" />

```

```

        <asp:ListItem Value="be" />
        <asp:ListItem Value="ca" />
        <asp:ListItem Value="de" />
        <asp:ListItem Value="dk" />
        <asp:ListItem Value="eu" />
        <asp:ListItem Value="fr" />
        <asp:ListItem Value="holl" />
        <asp:ListItem Value="it" />
        <asp:ListItem Value="se" />
        <asp:ListItem Value="uk" />
        <asp:ListItem Value="ukorg" />
        <asp:ListItem Value="us" />
    </asp:DropDownList>

    <asp:TextBox CssClass="move text2" ID="text12_Ursstad"
MaxLength="50" runat="server" Size="6"></asp:TextBox>

    <asp:DropDownList CssClass="move text2" ID="text13_Musiktyp"
runat="server" Size="1">
    <asp:ListItem />
    <asp:ListItem Value="Cyberpunk" />
    <asp:ListItem Value="EBM" />
    <asp:ListItem Value="EuroDisco" />
    <asp:ListItem Value="Gothrock" />
    <asp:ListItem Value="HipHop" />
    <asp:ListItem Value="Industri" />
    <asp:ListItem Value="New Wave" />
    <asp:ListItem Value="Pop" />
    <asp:ListItem Value="Punk" />
    <asp:ListItem Value="RnB" />
    <asp:ListItem Value="Rock" />
    <asp:ListItem Value="Synth" />
    <asp:ListItem Value="Synthpop" />
    </asp:DropDownList>

    <asp:TextBox CssClass="move text2" ID="text14_Kortbio"
MaxLength="6000" runat="server" Size="6"></asp:TextBox>

    <asp:TextBox CssClass="move text2" ID="text15_Inm_dat"
MaxLength="10"
        runat="server" Size="6"></asp:TextBox>

    <asp:RegularExpressionValidator id="vldDatum"
runat="server"
        Display="None"
        ValidationExpression="(19|20)\d\d[- /.](0[1-
9]|1[012])[- /.](0[1-9]|1[2][0-9]|3[01])"
        ControlToValidate="text15_Inm_dat" Enabled="true"
EnableClientScript="false"
        ValidationGroup="x" />

    <asp:DropDownList CssClass="move text2" ID="text16_Kop_grad"
runat="server" Size="1">
    <asp:ListItem />
    <asp:ListItem Value="1" />
    <asp:ListItem Value="2" />
    <asp:ListItem Value="3" />
    <asp:ListItem Value="4" />
    <asp:ListItem Value="5" />

```

```

</asp:DropDownList>

runat="server" <asp:DropDownList CssClass="move text2" ID="text17_Kop_kat"
Size="1">
  <asp:ListItem />
  <asp:ListItem Value="Ny" />
  <asp:ListItem Value="Beg" />
  <asp:ListItem Value="Rea" />
</asp:DropDownList>

MaxLength="4" <asp:TextBox CssClass="move text2" ID="text18_Ca_pris"
runat="server" Size="1"
></asp:TextBox>

runat="server" <asp:RegularExpressionValidator id="vldCaPris"
Display="None" ValidationExpression="^[0-9]{0,4}$"
ControlToValidate="text18_Ca_pris" Enabled="true"
EnableClientScript="false" ValidationGroup="x" />

<!--Knappar RAD1: -->

<asp:Button ID="button_visa_databas" runat="server"
CssClass="input move rad1" Text="Visa_TB"
onclick="button_visa_databas_Click"
CausesValidation="False" />

  <asp:Button ID="button_lagg_till" runat="server"
CssClass="input move rad1" Text="Lägg_till"
onclick="button_lagg_till_Click"
CausesValidation="true" ValidationGroup="x"/>

  <asp:Button ID="button_ta_bort" runat="server"
CssClass="input move rad1" Text="Ta_bort"
onclick="button_ta_bort_Click"
CausesValidation="False" />

  <asp:Button ID="button_uppdatera" runat="server"
CssClass="input move rad1" Text="Uppdatera"
onclick="button_uppdatera_Click"
CausesValidation="False" />

  <asp:Button ID="button_sok" runat="server"
CssClass="input move rad1" Text="Sök"
onclick="button_sok_Click"
CausesValidation="False" />

<!--Knappar - RAD2: -->

<asp:Button ID="btn21" runat="server"
CssClass="input move rad2" Text="Hämta"
onclick="button_fetch_Click"
CausesValidation="False"/>

<asp:Button ID="btn22" runat="server"

```

```

        CssClass="input move rad2" Text="Kommentar"
        onclick="button_Kommentar_Click"
        CausesValidation="False"/>

<asp:Button ID="btn23" runat="server"
    CssClass="input move rad2" Text="Överför"
    onclick="button_transfer_Click"
    CausesValidation="False"/>

<asp:Button ID="btn24" runat="server"
    CssClass="input move rad2" Text="Börja_om"
    OnClientClick="this.form.reset();return false;"
    onclick="button_Borja_om_Click"
    CausesValidation="False" />

    <asp:Button ID="btn25" runat="server"
    CssClass="input move rad2" Text="Omstart_TB"
    onclick="button_Refresh_TB_Click"
    CausesValidation="False" />

<!-- Tabellvals-radioknappar inkl. labels (i form av div): -->

<div class="label3 move" id="label13_TBskivor">Skivtabell</div>

    <asp:RadioButton ID="optSkivtabell" runat="server"
    GroupName="radioval" TextAlign="Right"
    AutoPostBack="true" CssClass="optSkivtabell" Checked="false"
    OnCheckedChanged="Skivtabell_CheckedChanged" />

    <div class="label3 move"
    id="label14_TBartist">Artisttabell</div>
    <asp:RadioButton ID="optArtisttabell" runat="server"
    GroupName="radioval" TextAlign="Right"
    AutoPostBack="true" CssClass="optArtisttabell"
    OnCheckedChanged="Artisttabell_CheckedChanged"
    Checked="false" />

    <div class="label3 move" id="label15_TBinkop">Inköpstabell</div>
    <asp:RadioButton ID="optInkopstabell" runat="server"
    GroupName="radioval" TextAlign="Right"
    AutoPostBack="true" CssClass="optInkopstabell"
    OnCheckedChanged="Inkopstabell_CheckedChanged"
    Checked="true" />

```

```
</div>
```

```
    </div>  
  </form>  
</body>  
</html>
```

Bakomliggande kod för Inkopstabellen där anrop till olika komponenter med olika funktioner sker. Mesta av felhanteringen sköts här och felmeddelanden ges till användaren när sådana har uppkommit.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Collections;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;
using DB_Callcode_Inkop.Inkop;

public partial class Default3 : System.Web.UI.Page
{
    //Instans för rensning av display och gridviews
    Clear_Inkop clr = new Clear_Inkop();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            //Initiering av range-värden för validering av artistnummer
            //vid tillägg av enhet(Lagg_till funktionen):
            Inkop aggz = new Inkop();
            int[] aggvalues = new int[2];
            aggvalues = aggz.GetAggCDNet();

            int minvalue = aggvalues[0];
            int maxvalue = aggvalues[1];

            vldArtistValidRange.MinimumValue = minvalue.ToString();

            if (maxvalue <= int.MaxValue)
            {
                vldArtistValidRange.MaximumValue = maxvalue.ToString();
            }
        }
    }

    //Knappar Rad 1:

    //Hämtar hela databasen:
    public void button_visa_databas_Click(object sender, EventArgs e)
    {
        //Leta upp kontroller:
```

```

        UserControl UGridArtister_Inkop_UCtrl =
(UserControl)Page.FindControl("UGridArtister_Inkop_UCtrl");

        GridView gridArtister_Inkop =
(GridView)UGridArtister_Inkop_UCtrl.FindControl("gridArtister_Inkop");

        Visa_databas vdb = new Visa_databas();

        vdb.button_visa_databas(gridArtister_Inkop, this.Page);
    }

    //Lägger till data i databas:
    protected void button_lagg_till_Click(object sender, EventArgs e)
    {
        try
        {
            Lagg_till lt = new Lagg_till();
            lt.button_lagg_till(display, text3_Artist, text4_Album,
text5_Format,
            text6_Press, text8_ar, text7_Kommentar, text15_Inm_dat,
text16_Kop_grad,
            text17_Kop_kat, text18_Ca_pris, vldDatum, vldCaPris,
vldArtistOnlyNum,
            vldArtistNotEmpty, vldArtistValidRange, this.Page);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);

            string mess = "<h2>[Page]Inkop.Lagg_till:
FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }

        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);

            string mess = "<h2>[Page]Inkop.Lagg_till:MySqlException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }

        catch (System.Exception err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);

            string mess =
"<h2>[Page]Inkop.Lagg_till:System.Exception</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
    }
}

```



```

//Tar bort data från databas:
protected void button_ta_bort_Click(object sender, EventArgs e)
{
    //Leta upp kontroller:
    GridView gridArtister_Inkop =
(GridView)UGridArtister_Inkop_UCtrl.FindControl("gridArtister_Inkop");
    GridView grdResult_Inkop =
(GridView)grdResult_Inkop_UCtrl.FindControl("grdResult_Inkop");

    try
    {
        Ta_bort tb = new Ta_bort();
        tb.button_ta_bort(gridArtister_Inkop, grdResult_Inkop,
            display, this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Ta_bort:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Ta_bort:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Ta_bort:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Uppdaterar databas:
protected void button_uppdatera_Click(object sender, EventArgs e)
{
    //Leta upp kontroller:
    GridView gridArtister_Inkop =
(GridView)UGridArtister_Inkop_UCtrl.FindControl("gridArtister_Inkop");

```

```

        GridView grdResult_Inkop =
(GridView)grdResult_Inkop_UCtrl.FindControl("grdResult_Inkop");

        try
        {
            Uppdatera upd = new Uppdatera();
            upd.button_oppdatera(gridArtister_Inkop, grdResult_Inkop,
            display, this.Page, text3_Artist, text4_Album, text5_Format,
            text6_Press, text8_ar, text7_Kommentar, text15_Inm_dat,
            text16_Kop_grad, text17_Kop_kat, text18_Ca_pris);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);

            string mess = "<h2>[Page]Inkop.Uppdatera:FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);

            string mess = "<h2>[Page]Inkop.Uppdatera:MySqlException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (System.Exception err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);

            string mess =
"<h2>[Page]Inkop.Uppdatera:System.Exception</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
    }

    //Söker i databas:
    protected void button_sok_Click(object sender, EventArgs e)
    {
        //Leta upp kontrollen:
        GridView grdResult_Inkop =
(GridView)grdResult_Inkop_UCtrl.FindControl("grdResult_Inkop");

        //Skapar instans av objekt för delayvärde och sök-anrop:
        Sok sk = new Sok();

        //Anger antal sekunder som sökningen skall vara i cache-objektet:
        sk.Delay = 20;

        try
        {

```

```

        sk.button_sok(grdResult_Inkop, this.Page,
        display, text3_Artist, text4_Album, text5_Format,
        text6_Press, text8_ar, text7_Kommentar, text15_Inm_dat,
        text16_Kop_grad, text17_Kop_kat, text18_Ca_pris);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Sok:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Sok:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Sok:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

//Knappar Rad 2:

```

//Hämtar en rad från aktuell tabell som skall uppdateras:
protected void button_fetch_Click(object sender, EventArgs e)
{
    //Leta upp kontroller:
    GridView gridArtister_Inkop =
    (GridView)UGridArtister_Inkop_UCtrl.FindControl("gridArtister_Inkop");
    GridView grdResult_Inkop =
    (GridView)grdResult_Inkop_UCtrl.FindControl("grdResult_Inkop");

    try
    {
        Fetch fch = new Fetch();
        fch.button_fetch(gridArtister_Inkop, grdResult_Inkop,

```

```

        display, display2, this.Page, text3_Artist, text4_Album,
text5_Format,
        text6_Press, text8_ar, text7_Kommentar, text15_Inm_dat,
text16_Kop_grad,
        text17_Kop_kat, text18_Ca_pris);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Fetch:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Fetch:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Fetch:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Visar panel för inmatning av längre kommentarer till Inkopstabell:
protected void button_Kommentar_Click(object sender, EventArgs e)
{
    //Leta upp kontroller:
    Panel pnlKommentar_Inkop =
    (Panel)pnlKommentar_Inkop_UCtrl.FindControl("pnlKommentar_Inkop");

    try
    {
        Visa_KommentarPanel kpnl = new Visa_KommentarPanel();
        kpnl.button_kommentarpanel(pnlKommentar_Inkop, this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Kommentar:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

```

    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Kommentar:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[Page]Inkop.Kommentar:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

```

//Överför vald post från Inkopstabell till skivtabell:
protected void button_transfer_Click(object sender, EventArgs e)
{
    //Leta upp kontroller:
    GridView gridArtister_Inkop =
(Gridview)UGridArtister_Inkop_UCtrl.FindControl("gridArtister_Inkop");
    GridView grdResult_Inkop =
(Gridview)grdResult_Inkop_UCtrl.FindControl("grdResult_Inkop");

    try
    {
        Transfer trf = new Transfer();
        trf.button_transfer(gridArtister_Inkop, grdResult_Inkop,
            display, this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Transfer:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Transfer:MySqlException</h2>";
    }
}

```

```

        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Transfer:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Raderar innehållet i alla textrutor och dropdownmenyer:
protected void button_Borja_om_Click(object sender, EventArgs e)
{
    Borja_om bro = new Borja_om();
    bro.button_Borja_om();
}

//Fungerar som en omstart och rensar alla inmatningsrutor odyl +
display:
protected void button_Refresh_TB_Click(object sender, EventArgs e)
{
    HttpServerUtility server = this.Server;
    String retursida = "~/Default3.aspx";

    try
    {
        Refresh_TB rtb = new Refresh_TB();
        rtb.button_Refresh_TB(server, retursida);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[Page]Inkop.Refresh_TB:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess = "<h2>[Page]Inkop.Refresh_TB:MySqlException</h2>";

```

```

        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[Page] Inkop.Refresh_TB: System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

//Växla mellan tabeller:

```

protected void Skivtabell_CheckedChanged(object sender, EventArgs e)
{
    IDictionaryEnumerator CacheEnum =
HttpRuntime.Cache.GetEnumerator();
    while (CacheEnum.MoveNext())
    {
        string key = CacheEnum.Key.ToString();
        HttpRuntime.Cache.Remove(key);
    }
    Server.Transfer("~/Default.aspx");
}

protected void Artisttabell_CheckedChanged(object sender, EventArgs e)
{
    IDictionaryEnumerator CacheEnum =
HttpRuntime.Cache.GetEnumerator();
    while (CacheEnum.MoveNext())
    {
        string key = CacheEnum.Key.ToString();
        HttpRuntime.Cache.Remove(key);
    }
    Server.Transfer("~/Default2.aspx");
}

protected void Inkoptabell_CheckedChanged(object sender, EventArgs e)
{
    IDictionaryEnumerator CacheEnum =
HttpRuntime.Cache.GetEnumerator();
    while (CacheEnum.MoveNext())
    {
        string key = CacheEnum.Key.ToString();
        HttpRuntime.Cache.Remove(key);
    }
}

```

```
        Server.Transfer("~/Default3.aspx");  
    }  
  
}
```


Frontend - Usercontrols; GridViews och paneler i par med tillhörande cs-filer

Bildar struktur för varje aktuell Usercontrol och i likhet apx-filerna ovan även med tillhörande bakomliggande kod, också här sker listningen parvis.

Strukturen för gridview som visar artistdata (snabbdata om viss artist) gällande Skivtabellen.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="grd_Artistdata.ascx.cs" Inherits="UserControls_grd_Artistdata" %>

<!-- GridView som visar data om en artist: -->

<asp:GridView ID="grd_Artistdata" runat="server" AllowPaging="false"
    OnRowCommand="grd_Artistdata_RowCommand" HeaderStyle-
    CssClass="header">

    <Columns>

        <asp:BoundField DataField="Artist" HeaderText="Artist" ItemStyle-
        CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="VPfr" HeaderText="VPfr" ItemStyle-
        CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="Vptill" HeaderText="Vptill"
        ItemStyle-CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="Ursland" HeaderText="Ursland"
        ItemStyle-CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="Ursstad" HeaderText="Ursstad"
        ItemStyle-CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="Musiktyp" HeaderText="Musiktyp"
        ItemStyle-CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:TemplateField HeaderText="Kort biografi" ItemStyle-
        CssClass="grid_style"
            HeaderStyle-CssClass="pos_kortbio">
            <ItemTemplate>
                <asp:Button ID="Artist_data" runat="server"
                CssClass="arti" Text='<%#Eval("Artist") %>'
                CommandArgument='<%#Eval("Artist") %>'
                CommandName="Artist_kortbio" />
            </ItemTemplate>
        </asp:TemplateField>

    </Columns>

</asp:GridView>
```

Bakomliggande kod till strukturen för artistdata till Skivtabellen. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

using DB_Callcode.Skivor;

public partial class UserControls_grd_Artistdata :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear clr = new Clear();

    protected void Page_Load(object sender, EventArgs e){}

    //Visar en artists kortbiografi:
    protected void grd_Artistdata_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        //Leta upp kontrollen:
        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

        try
        {
            dbtn_Visa_Kortbio vkb = new dbtn_Visa_Kortbio();
            vkb.grd_Artistdata_RowCommand(display, e, this.Page);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            string mess =
"<h2>[Uctrl]UCtrlgrdArtistdata.dbtn_Visa_Kortbio:FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            string mess =
"<h2>[Uctrl]UCtrlgrdArtistdata.dbtn_Visa_Kortbio:MySqlException</h2>";
```

```

        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Uctrl]UCtrlgrdArtistdata.dbtn_Visa_Kortbio:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}
}

```

Strukturen för gridView som visar artistdata (snabbdata om viss artist) gällande Inkopstabellen.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="grd_Artistdata_Inkop.ascx.cs"
Inherits="UserControls_grd_Artistdata_Inkop" %>

<!-- GridView som visar data om en artist: -->

<asp:GridView ID="grd_Artistdata_Inkop" runat="server"
AllowPaging="false"
    OnRowCommand="grd_Artistdata_Inkop_RowCommand" HeaderStyle-
CssClass="header">

    <Columns>

        <asp:BoundField DataField="Artist" HeaderText="Artist" ItemStyle-
CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="VPfr" HeaderText="VPfr" ItemStyle-
CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="Vptill" HeaderText="Vptill" ItemStyle-
CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="Ursland" HeaderText="Ursland" ItemStyle-
CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="Ursstad" HeaderText="Ursstad" ItemStyle-
CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:BoundField DataField="Musiktyp" HeaderText="Musiktyp" ItemStyle-
CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartistdata" />

        <asp:TemplateField HeaderText="Kort biografi" ItemStyle-
CssClass="grid_style"
            HeaderStyle-CssClass="pos_kortbio" >
            <ItemTemplate>
                <asp:Button ID="Artist_data" runat="server" CssClass="arti"
Text='<%#Eval("Artist") %>'
                    CommandArgument='<%#Eval("Artist") %>'
CommandName="Artist_kortbio" />
            </ItemTemplate>
        </asp:TemplateField>

    </Columns>

</asp:GridView>
```

Bakomliggande kod till strukturen för artistdata till Inkopstabellen. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

using DB_Callcode_Inkop.Inkop;

public partial class UserControls_grd_Artistdata_Inkop :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear_Inkop clr = new Clear_Inkop();

    protected void Page_Load(object sender, EventArgs e) { }

    //Visar en artists kortbiografi:
    protected void grd_Artistdata_Inkop_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        //Leta upp kontroller:
        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

        try
        {
            dbtn_Visa_Kortbio vkb = new dbtn_Visa_Kortbio();
            vkb.grd_Artistdata_Inkop_RowCommand(display, e, this.Page);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);

            string mess =
"<h2>[Uctrl]UctrlgrdArtistdata_Inkop.dbtn_Visa_Kortbio:FormatException</h2>";

            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);
        }
    }
}
```

```

        string mess =
"<h2>[Uctrl]UCtrlgrdArtistdata_Inkop.dbtn_Visa_Kortbio:MySqlException</h2>"
;
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[Uctrl]UCtrlgrdArtistdata_Inkop.dbtn_Visa_Kortbio:System.Exception</h2>"
>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}
}

```

Strukturen för gridview som visar resultat från Fetch (hämtad data för uppdatering) och sökning gällande Skivtabellen.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="GridView_grdResult.ascx.cs"
Inherits="UserControls_Gridview_grdResult" %>

<!-- GridView för att visa resultat från Fetch (hämtad data för
uppdatering) och sökning: -->

<asp:GridView ID="grdResult" runat="server" BackColor="White"
AllowPaging="true" PageSize="6"
    AllowSorting="false" OnRowCommand="gridResult_RowCommand"
    HeaderStyle-CssClass="header"
OnPageIndexChanging="grdResult_PageIndexChanging"
    AutoGenerateColumns="false">

    <Columns>

        <asp:TemplateField HeaderText="Mark" ItemStyle-CssClass="grid_style"
HeaderStyle-CssClass="pos_grdresult" >
            <HeaderTemplate>

                <TF:DerivCheckBox ID="chkCDval_ALL" runat="server"
Checked='<%# Bind("Val") %>' AutoPostBack="true"
                UnText='<%#Eval("#") %>'
OnCheckedChanged="chkCDvalALLGrdRes_CheckedChanged" />

            </HeaderTemplate>

            <ItemTemplate>

                <TF:DerivCheckBox ID="chkCDval" runat="server" Checked='<%#
Bind("Val") %>' AutoPostBack="true"
                UnText='<%#Eval("#") %>'
OnCheckedChanged="chkCDval_CheckedChanged" />

            </ItemTemplate>

        </asp:TemplateField>

        <asp:TemplateField ItemStyle-CssClass="grid_style"
SortExpression="Artist" HeaderStyle-CssClass="pos_grdresult">

            <HeaderTemplate>
                <asp:LinkButton ID="lnkbtnArtist" runat="server"
CommandName="Sort"
                ForeColor="#F7F7F7" CommandArgument="Artist"
OnCommand="SortSkivor_Command">
                    Artist</asp:LinkButton>
            </HeaderTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
```



```

        </HeaderTemplate>
        <ItemTemplate>
            <asp:Button ID="Artist" runat="server" CssClass="arti"
Text='<%#Eval("Artist") %>'
            CommandArgument='<%#Eval("Artist_no") %>' CommandName="Artist"
            OnCommand="SortSkivor_Command" />
        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

```

```

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnAlbum" runat="server"
CommandName="Sort"
            ForeColor="#F7F7F7" CommandArgument="Album"
OnCommand="SortSkivor_Command">
                Titel</asp:LinkButton>
            </HeaderTemplate>
        <ItemTemplate>
            <asp:Label ID="lblSortArtist" runat="server" Text='<%#
Eval("Album") %>'
                EnableTheming="false">
            </asp:Label>
        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

```

```

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnFormat" runat="server"
CommandName="Sort"
            ForeColor="#F7F7F7" CommandArgument="Format"
OnCommand="SortSkivor_Command">
                Format</asp:LinkButton>
            </HeaderTemplate>
        <ItemTemplate>
            <asp:Label ID="lblSortFormat" runat="server" Text='<%#
Eval("Format") %>'
                EnableTheming="false">
            </asp:Label>
        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

```

```

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnPress" runat="server"
CommandName="Sort"
            ForeColor="#F7F7F7" CommandArgument="Press"
OnCommand="SortSkivor_Command">
                Press</asp:LinkButton>
            </HeaderTemplate>
        <ItemTemplate>

```

```

        <asp:Label ID="lblSortPress" runat="server" Text='<%#
Eval("Press") %>'
        EnableTheming="false">
    </asp:Label>
</ItemTemplate>
<ItemStyle CssClass="grid_style" />
</asp:TemplateField>

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnAr" runat="server" CommandName="Sort"
                ForeColor="#F7F7F7" CommandArgument="Ar"
OnCommand="SortSkivor_Command">
                År</asp:LinkButton>
        </HeaderTemplate>
        <ItemTemplate>
            <asp:Label ID="lblSortAr" runat="server" Text='<%# Eval("Ar") %>'
                EnableTheming="false">
            </asp:Label>
        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

    <asp:TemplateField HeaderText="Komtr" HeaderStyle-CssClass="cmt_align"
        ItemStyle-CssClass="grid_style">
        <ItemTemplate>
            <asp:Button ID="kommentar" runat="server" CssClass="cmt"
Text='<%#Eval("#") %>'
                CommandArgument='<%#Eval("#") %>' CommandName="Kommentar" />
        </ItemTemplate>
        <HeaderStyle CssClass="cmt_align" />
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

</Columns>

</asp:GridView>

```

Bakomliggande kod till strukturen för gridview som visar resultat från Fetch (hämtad data för uppdatering) och sökning gällande Skivtabellen. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Collections;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;
using DB_Callcode.Skivor;

public partial class UserControls_Gridview_grdResult :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear clr = new Clear();

    protected void Page_Load(object sender, EventArgs e){}

    //Kommentarknapp och Artistknapp:
    protected void gridResult_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        //Leta upp kontroller:
        UserControl grd_Artistdata_UCtrl =
(UserControl)Page.FindControl("grd_Artistdata_UCtrl");

        GridView grd_Artistdata =
(GridView)grd_Artistdata_UCtrl.FindControl("grd_Artistdata");

        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");
        HtmlGenericControl display2 =
(HtmlGenericControl)Page.FindControl("display2");

        try
        {
            dbtn_KomArt kart = new dbtn_KomArt();
            kart.gridArtister_RowCommand(grd_Artistdata, display, display2,
this.Page, e);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);
        }
    }
}
```

```

        string mess =
"<h2>[Uctrl]UserCgrdResult.dbtn_KomArt:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Uctrl]UserCgrdResult.dbtn_KomArt:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        String mess =
"<h2>[Uctrl]UserCgrdResult.dbtn_KomArt:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Kryssruta för gridview gridArtister och grdResult:
protected void chkCDval_CheckedChanged(object sender, EventArgs e)
{
    //Leta upp kontroller:

    UserControl gridArtister_Uctrl =
(UserControl)Page.FindControl("gridArtister_Uctrl");

    GridView gridArtister =
(GridView)gridArtister_Uctrl.FindControl("gridArtister");

    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        chk_Gridviews cgrd = new chk_Gridviews();
        cgrd.chkCDval_CheckedChanged(gridArtister, grdResult);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[Uctrl]UserCgrdResult.chk_Gridviews:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews

```

```

        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult.chk_Gridviews:MySQLException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult.chk_Gridviews:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Headerkryssruta för gridview grdResult:
public void chkCDvalALLGrdRes_CheckedChanged(object sender, EventArgs
e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        hchk_grdResult hchk = new hchk_grdResult();
        hchk.chkCDvalALLGrdRes_CheckedChanged(grdResult, sender);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult.hchk_grdResult:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySQLException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult.hchk_grdResult:MySQLException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);
    }
}

```

```

        String mess =
"<h2>[UCtrl]UserCgrdResult.hchk_grdResult:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }

}

//Sortering för gridview som är programatiskt ansluten till databas
protected void SortSkivor_Command(object sender, CommandEventArgs e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    DataSet ds = new DataSet();
    DataView dv = new DataView();

    //Hämta värden så cachekeys kan bli likadana som de vid sökning,
    //dvs individuella cachekeys
    TextBox artist = (TextBox)Page.FindControl("text3_Artist");
    TextBox album = (TextBox)Page.FindControl("text4_Album");
    DropDownList format =
(DropDownList)Page.FindControl("text5_Format");
    DropDownList press = (DropDownList)Page.FindControl("text6_Press");
    DropDownList ar = (DropDownList)Page.FindControl("text8_ar");
    TextBox kommentar = (TextBox)Page.FindControl("text7_Kommentar");

    ds = (DataSet)HttpRuntime.Cache.Get("cache" + artist.Text +
album.Text
        + format.Text + press.Text + ar.Text +
kommentar.Text);

    //Första tabellen i datasetet tilldelas till en dataview:
    if (this.Cache["cache" + artist.Text + album.Text
        + format.Text + press.Text + ar.Text +
kommentar.Text] == null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        display.InnerHtml = "<h2>Cacheminnets delaytid har gått ut.
Genomför en ny sökning</h2>";
    }
    else
        dv = ds.Tables[0].DefaultView;
}

```

```

        //Kollar om rad är sorterbar, vilken rad som i så fall är
sorterbar, och om
        //den skall sorteras mha stigande eller fallande sorteringsordning:
        if (e.CommandName.Equals("Sort"))
        {
            if (this.ViewState["SortExp"] == null)
            {
                this.ViewState["SortExp"] = e.CommandArgument.ToString();
                this.ViewState["SortOrder"] = "ASC";
            }
            else
            {
                if (this.ViewState["SortExp"].ToString() ==
e.CommandArgument.ToString())
                {
                    if (this.ViewState["SortOrder"].ToString() == "ASC")
                        this.ViewState["SortOrder"] = "DESC";
                    else
                        this.ViewState["SortOrder"] = "ASC";
                }
                else
                {
                    this.ViewState["SortOrder"] = "ASC";
                    this.ViewState["SortExp"] =
e.CommandArgument.ToString();
                }
            }
        }

        //Om 'SortExp' (aktuell kolumnrad) är tillgänglig så sorteras
        //den mha dataview enligt inställningarna ovan:
        if (this.ViewState["SortExp"] != null)
        {
            string errmess = "<h2>Ingen tabell i DataView pga
cachminnets ";
            errmess += "delaytid har gått ut. Genomför en ny
sökning</h2>";

            if (dv.Table == null)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(this.Page);

                display.InnerHtml = errmess;
            }
            else
            {
                dv.Sort = this.ViewState["SortExp"].ToString()
                    + " " +
this.ViewState["SortOrder"].ToString();
            }

            //Skapar nytt objekt och adderar sorterade dataview:n till det:
            GridView gr = new GridView();

            //CurrentPageIndex:
            grdResult.DataSource = dv;
            grdResult.DataBind();
            grdResult.Visible = true;

```

```

    }
}

//Paging för gridview som är programatiskt ansluten till databas
protected void grdResult_PageIndexChanging(object sender,
GridViewPageEventArgs e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    DataSet ds = new DataSet();
    DataView dv = new DataView();

    //Hämta värden så cachekeys kan bli likadana som de vid sökning,
    //dvs individuella cachekeys
    TextBox artist = (TextBox)Page.FindControl("text3_Artist");
    TextBox album = (TextBox)Page.FindControl("text4_Album");
    DropDownList format =
(DropDownList)Page.FindControl("text5_Format");
    DropDownList press = (DropDownList)Page.FindControl("text6_Press");
    DropDownList ar = (DropDownList)Page.FindControl("text8_ar");
    TextBox kommentar = (TextBox)Page.FindControl("text7_Kommentar");

    ds = (DataSet)HttpRuntime.Cache.Get("cache" + artist.Text +
album.Text
+ format.Text + press.Text + ar.Text +
kommentar.Text);

    //Första tabellen i datasetet tilldelas till en dataview:
    if (this.Cache["cache" + artist.Text + album.Text
+ format.Text + press.Text + ar.Text +
kommentar.Text] == null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        display.InnerHtml = "<h2>Cacheminnets delaytid har gått ut.
Genomför en ny sökning</h2>";
    }
    else
        dv = ds.Tables[0].DefaultView;

    //Om 'SortExp' (aktuell kolumnrad) är tillgänglig så sorteras
    //den mha dataview enligt värdena i viewstatevariablerna:
    if (this.ViewState["SortExp"] != null)
    {
        string errmess = "<h2>Ingen tabell i DataView pga cachminnets
";
        errmess += "delaytid har gått ut. Genomför en ny sökning</h2>";
    }
}

```



```

        if (dv.Table == null)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            display.InnerHtml = errmess;
        }
        else
        {
            dv.Sort = this.ViewState["SortExp"].ToString()
                + " " + this.ViewState["SortOrder"].ToString();
        }
    }

    //Skapar nytt objekt och ett nytt sidindex så man kan bläddra
    (paging)
    GridView gr = new GridView();

    gr = (GridView)sender;
    gr = grdResult;

    RememberOldValues();

    gr.PageIndex = e.NewPageIndex;

    //CurrentPageIndex
    grdResult.DataSource = dv;
    grdResult.DataBind();

    RePopulateValues();

    grdResult.Visible = true;
}

//Sparar kryssrutors state-värden mha bool och Session-objekt:
protected void RememberOldValues()
{
    //Lika många utrymmen för kryssrytevärden som antal rader/sid:
    bool[] values = new bool[grdResult.PageSize];
    DerivCheckBox chb;

    //Hittar kryssruta och sparar dess statevärde i array:
    for (int i = 0; i < grdResult.Rows.Count; i++)
    {
        chb = (DerivCheckBox)grdResult.Rows[i].FindControl("chkCDval");
        if (chb != null)
        {
            values[i] = chb.Checked;
        }
    }
}

//Sparar bool-array i Session för senare access:

```

```

        Session["page" + grdResult.PageIndex] = values;
    }

    //Återskapar bool-array från Sessionobjekt och dess statevärden
    //för kryssrutorna på aktuell sida:
    protected void RePopulateValues()
    {
        if (Session["page" + grdResult.PageIndex] != null)
        {
            DerivCheckBox chb;
            bool[] values = (bool[])Session["page" + grdResult.PageIndex];
            for (int i = 0; i < grdResult.Rows.Count; i++)
            {
                chb =
                (DerivCheckBox)grdResult.Rows[i].FindControl("chkCDval");
                chb.Checked = values[i];
                if (chb.Checked == true)
                {
                    grdResult.Rows[i].BackColor =
                    System.Drawing.ColorTranslator.FromHtml("#738A9C");
                    grdResult.Rows[i].ForeColor =
                    System.Drawing.ColorTranslator.FromHtml("#F7F7F7");
                    grdResult.Rows[i].Font.Bold = true;
                }
            }
        }
    }
}

```

Strukturen för gridview som visar resultat från Fetch (hämtad data för uppdatering) och sökning gällande Artisttabellen.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Gridview_grdResult_Artist.ascx.cs"
Inherits="UserControls_Gridview_grdResult_Artist" %>

<!-- GridView för att visa resultat från Fetch (hämtad data för
uppdatering) och sökning: -->

<asp:GridView ID="grdResult_Artist" runat="server" BackColor="White"
AllowPaging="true" PageSize="6"
    AllowSorting="false" OnRowCommand="gridResult_RowCommand_Artist"
HeaderStyle-CssClass="header"
    OnPageIndexChanging="grdResultArtist_PageIndexChanging"
AutoGenerateColumns="false">

    <RowStyle Height="35px" />

    <Columns>

        <asp:TemplateField HeaderText="Mark" ItemStyle-CssClass="grid_style"
HeaderStyle-CssClass="pos_gridartist">
            <HeaderTemplate>

                <TF:DerivCheckBox ID="chkArtval_ALL" runat="server"
Checked='<%# Bind("ValArt") %>' AutoPostBack="true"
                UnText='<%#Eval("No") %>'
OnCheckedChanged="chkArtvalALLGrdRes_CheckedChanged" />

            </HeaderTemplate>

            <ItemTemplate>

                <TF:DerivCheckBox ID="chkArtval" runat="server" Checked='<%#
Bind("ValArt") %>' AutoPostBack="true"
                UnText='<%#Eval("No") %>'
OnCheckedChanged="chkArtval_CheckedChanged" />

            </ItemTemplate>

        </asp:TemplateField>

        <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
            <HeaderTemplate>
                <asp:LinkButton ID="lnkbtnNo" runat="server" CommandName="Sort"
ForeColor="#F7F7F7" CommandArgument="No"
OnCommand="SortArtist_Command">
                    ArtNo</asp:LinkButton>
            </HeaderTemplate>
```

```

        <ItemTemplate>
            <asp:Label ID="lblSortNo" runat="server" Text='<%# Eval("No") %>'
                EnableTheming="false">
            </asp:Label>
        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

    <asp:TemplateField ItemStyle-CssClass="grid_style"
        SortExpression="Artist" HeaderStyle-CssClass="pos_gridartist">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnArtist" runat="server"
CommandName="Sort"
                ForeColor="#F7F7F7" CommandArgument="Artist"
OnCommand="SortArtist_Command">
                Artist</asp:LinkButton>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:Button ID="Artist" runat="server" CssClass="arti"
Text='<%#Eval("Artist") %>'
                CommandArgument='<%#Eval("Artist") %>' CommandName="Artist_kortbio" />
            </ItemTemplate>
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnVPfr" runat="server" CommandName="Sort"
                ForeColor="#F7F7F7" CommandArgument="VPfr"
OnCommand="SortArtist_Command">
                VPfr</asp:LinkButton>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:Label ID="lblSortVPfr" runat="server" Text='<%# Eval("VPfr")
%>'
                EnableTheming="false">
            </asp:Label>
        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnVPtill" runat="server"
CommandName="Sort"
                ForeColor="#F7F7F7" CommandArgument="VPtill"
OnCommand="SortArtist_Command">
                VPtill</asp:LinkButton>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:Label ID="lblSortVPtill" runat="server" Text='<%#
Eval("VPtill") %>'
                EnableTheming="false">
            </asp:Label>
        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

```

```

        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
    CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnUrsland" runat="server"
    CommandName="Sort"
            ForeColor="#F7F7F7" CommandArgument="Ursland"
    OnCommand="SortArtist_Command">
                Ursland</asp:LinkButton>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:Label ID="lblSortUrsland" runat="server" Text='<%#
Eval("Ursland") %>'
                EnableTheming="false">
            </asp:Label>
            </ItemTemplate>
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
    CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnUrsstad" runat="server"
    CommandName="Sort"
            ForeColor="#F7F7F7" CommandArgument="Ursstad"
    OnCommand="SortArtist_Command">
                Ursstad</asp:LinkButton>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:Label ID="lblSortUrsstad" runat="server" Text='<%#
Eval("Ursstad") %>'
                EnableTheming="false">
            </asp:Label>
            </ItemTemplate>
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
    CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnMusiktyp" runat="server"
    CommandName="Sort"
            ForeColor="#F7F7F7" CommandArgument="Musiktyp"
    OnCommand="SortArtist_Command">
                Musiktyp</asp:LinkButton>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:Label ID="lblSortMusiktyp" runat="server" Text='<%#
Eval("Musiktyp") %>'
                EnableTheming="false">
            </asp:Label>
            </ItemTemplate>
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

```

```
</Columns>  
</asp:GridView>
```

Bakomliggande kod till strukturen för gridview som visar resultat från Fetch (hämtad data för uppdatering) och sökning gällande Artisttabellen. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

using DB_Callcode_Artist.Artist;

public partial class UserControl_Gridview_grdResult_Artist :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear_Artist clr = new Clear_Artist();

    protected void Page_Load(object sender, EventArgs e) { }

    //Artistknapp:
    protected void gridResult_RowCommand_Artist(object sender,
GridViewCommandEventArgs e)
    {
        //Leta upp kontroller:

        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

        try
        {
            dbtn_Visa_Kortbio vkb = new dbtn_Visa_Kortbio();
            vkb.grd_Artistdata_Artist_RowCommand(display, e, this.Page);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);

            string mess =
"<h2>[Uctrl]UserCgrdResult_Artist.dbtn_KomArt:FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);
        }
    }
}
```

```

        string mess =
"<h2>[UCtrl]UserCgrdResult_Artist.dbtn_KomArt:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        String mess =
"<h2>[UCtrl]UserCgrdResult_Artist.dbtn_KomArt:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Kryssruta för gridview gridArtister_Artist och grdResult_Artist:
protected void chkArtval_CheckedChanged(object sender, EventArgs e)
{
    //Leta upp kontroller:

    UserControl gridArtister_Artist_UCtrl =
(UserControl)Page.FindControl("gridArtister_Artist_UCtrl");
    UserControl pnlKortbio_UCtrl =
(UserControl)Page.FindControl("pnlKortbio_UCtrl");

    GridView gridArtister_Artist =
(GridView)gridArtister_Artist_UCtrl.FindControl("gridArtister_Artist");
    Panel pnlKortbio =
(Panel)pnlKortbio_UCtrl.FindControl("pnlKortbio");

    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        chk_Gridviews cgrd = new chk_Gridviews();
        cgrd.chkArtval_CheckedChanged(gridArtister_Artist,
grdResult_Artist);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult_Artist.chk_Gridviews:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);
    }
}

```



```

        string mess =
"<h2>[UCtrl]UserCgrdResult_Artist.chk_Gridviews:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult_Artist.chk_Gridviews:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Headerkryssruta för gridview grdResult_Artist:
protected void chkArtvalALLGrdRes_CheckedChanged(object sender,
EventArgs e)
{
    //Leta upp kontroller:
    UserControl grdResult_Artist_UCtrl =
(UserControl)Page.FindControl("grdResult_Artist_UCtrl");

    GridView grdResult_Artist =
(GridView)grdResult_Artist_UCtrl.FindControl("grdResult_Artist");

    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        hchk_grdResult hchk = new hchk_grdResult();
        hchk.chkArtvalALLGrdRes_CheckedChanged(grdResult_Artist,
sender);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult_Artist.hchk_grdResult:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult_Artist.hchk_grdResult:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)

```

```

    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        String mess =
"<h2>[UCtrl]UserCgrdResult_Artist.hchk_grdResult:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Sortering för gridview som är programatiskt ansluten till databas
protected void SortArtist_Command(object sender, CommandEventArgs e)
{
    //Leta upp kontroller:
    UserControl grdResult_Artist_UCtrl =
(UserControl)Page.FindControl("grdResult_Artist_UCtrl");

    GridView grdResult_Artist =
(GridView)grdResult_Artist_UCtrl.FindControl("grdResult_Artist");

    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    DataSet ds = new DataSet();
    DataView dv = new DataView();

    //Hämta värden så cachekeys kan bli likadana som de vid sökning,
    //dvs individuella cachekeys
    TextBox artist = (TextBox)Page.FindControl("text3_Artist");
    DropDownList vpfr = (DropDownList)Page.FindControl("text9_VPfr");
    DropDownList vptill =
(DropDownList)Page.FindControl("text10_VPtill");
    DropDownList ursland =
(DropDownList)Page.FindControl("text11_Ursland");
    TextBox ursstad = (TextBox)Page.FindControl("text12_Ursstad");
    DropDownList musiktyp =
(DropDownList)Page.FindControl("text13_Musiktyp");

    ds = (DataSet)HttpRuntime.Cache.Get("cache" + artist.Text +
vpfr.Text + vptill.Text + ursland.Text
        + ursstad.Text + musiktyp.Text);

    //Första tabellen i datasetet tilldelas till en dataview:
    if (this.Cache["cache" + artist.Text + vpfr.Text + vptill.Text +
ursland.Text
        + ursstad.Text + musiktyp.Text] == null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        display.InnerHtml = "<h2>Cacheminnets delaytid har gått ut.
Genomför en ny sökning</h2>";
    }
    else

```

```

dv = ds.Tables[0].DefaultView;

//Kollar om rad är sorterbar, vilken rad som i så fall är
sorterbar, och om
//den skall sorteras mha stigande eller fallande sorteringsordning:
if (e.CommandName.Equals("Sort"))
{
    if (this.ViewState["SortExp"] == null)
    {
        this.ViewState["SortExp"] = e.CommandArgument.ToString();
        this.ViewState["SortOrder"] = "ASC";
    }
    else
    {
        if (this.ViewState["SortExp"].ToString() ==
e.CommandArgument.ToString())
        {
            if (this.ViewState["SortOrder"].ToString() == "ASC")
                this.ViewState["SortOrder"] = "DESC";
            else
                this.ViewState["SortOrder"] = "ASC";
        }
        else
        {
            this.ViewState["SortOrder"] = "ASC";
            this.ViewState["SortExp"] =
e.CommandArgument.ToString();
        }
    }
}

//Om 'SortExp' (aktuell kolumnrad) är tillgänglig så sorteras
//den mha dataview enligt inställningarna ovan:
if (this.ViewState["SortExp"] != null)
{
    string errmess = "<h2>Ingen tabell i DataView pga
cachminnets ";
    errmess += "delaytid har gått ut. Genomför en ny
sökning</h2>";

    if (dv.Table == null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        display.InnerHtml = errmess;
    }
    else
    {
        dv.Sort = this.ViewState["SortExp"].ToString()
            + " " +
this.ViewState["SortOrder"].ToString();
    }

    //Skapar nytt objekt och adderar sorterade dataview:n till det:
    GridView gr = new GridView();

    //CurrentPageIndex:

```

```

        grdResult_Artist.DataSource = dv;
        grdResult_Artist.DataBind();
        grdResult_Artist.Visible = true;
    }
}

//Paging för gridview som är programatiskt ansluten till databas
protected void grdResultArtist_PageIndexChanging(object sender,
GridViewPageEventArgs e)
{
    //Leta upp kontroller:
    UserControl grdResult_Artist_UCtrl =
(UserControl)Page.FindControl("grdResult_Artist_UCtrl");

    GridView grdResult_Artist =
(GridView)grdResult_Artist_UCtrl.FindControl("grdResult_Artist");

    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    DataSet ds = new DataSet();
    DataView dv = new DataView();

    //Hämta värden så cachekeys kan bli likadana som de vid sökning,
    //dvs individuella cachekeys
    TextBox artist = (TextBox)Page.FindControl("text3_Artist");
    DropDownList vpfr = (DropDownList)Page.FindControl("text9_VPfr");
    DropDownList vptill =
(DropDownList)Page.FindControl("text10_VPtill");
    DropDownList ursland =
(DropDownList)Page.FindControl("text11_Ursland");
    TextBox ursstad = (TextBox)Page.FindControl("text12_Ursstad");
    DropDownList musiktyp =
(DropDownList)Page.FindControl("text13_Musiktyp");

    ds = (DataSet)HttpRuntime.Cache.Get("cache" + artist.Text +
vpfr.Text + vptill.Text + ursland.Text
+ ursstad.Text + musiktyp.Text);

    //Första tabellen i datasetet tilldelas till en dataview:
    if (this.Cache["cache" + artist.Text + vpfr.Text + vptill.Text +
ursland.Text
+ ursstad.Text + musiktyp.Text] == null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        display.InnerHtml = "<h2>Cacheminnets delaytid har gått ut.
Genomför en ny sökning</h2>";
    }
}

```

```

    }
    else
        dv = ds.Tables[0].DefaultView;

    //Om 'SortExp' (aktuell kolumnrad) är tillgänglig så sorteras
    //den mha dataview enligt värdena i viewstatevariablerna:
    if (this.ViewState["SortExp"] != null)
    {
        string errmess= "<h2>Ingen tabell i DataView pga cachminnets ";
        errmess += "delaytid har gått ut. Genomför en ny sökning</h2>";

        if (dv.Table == null)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);

            display.InnerHtml = errmess;
        }
        else
            dv.Sort = this.ViewState["SortExp"].ToString()
                + " " + this.ViewState["SortOrder"].ToString();
    }

    //Skapar nytt objekt och ett nytt sidindex så man kan bläddra
    (paging) GridView gr = new GridView();

    gr = (GridView)sender;
    gr = grdResult_Artist;

    RememberOldValues();

    gr.PageIndex = e.NewPageIndex;

    //CurrentPageIndex
    grdResult_Artist.DataSource = dv;
    grdResult_Artist.DataBind();

    RePopulateValues();

    grdResult_Artist.Visible = true;
}

//Sparar kryssrutors state-värden mha bool och Session-objekt:
protected void RememberOldValues()
{
    //Lika många utrymmen för kryssrytevärden som antal rader/sid:
    bool[] values = new bool[grdResult_Artist.PageSize];
    DerivCheckBox chb;

    //Hittar kryssruta och sparar dess statevärde i array:
    for (int i = 0; i < grdResult_Artist.Rows.Count; i++)

```

```

        {
            chb =
(DerivCheckBox)grdResult_Artist.Rows[i].FindControl("chkArtval");
            if (chb != null)
            {
                values[i] = chb.Checked;
            }
        }
        //Sparar bool-array i Session för senare access:
        Session["page" + grdResult_Artist.PageIndex] = values;
    }

    //Återskapar bool-array från Sessionobjekt och dess statevärden
    //för kryssrutorna på aktuell sida:
    protected void RePopulateValues()
    {
        if (Session["page" + grdResult_Artist.PageIndex] != null)
        {
            DerivCheckBox chb;
            bool[] values = (bool[])Session["page" +
grdResult_Artist.PageIndex];
            for (int i = 0; i < grdResult_Artist.Rows.Count; i++)
            {
                chb =
(DerivCheckBox)grdResult_Artist.Rows[i].FindControl("chkArtval");
                chb.Checked = values[i];
                if (chb.Checked == true)
                {
                    grdResult_Artist.Rows[i].BackColor =
System.Drawing.ColorTranslator.FromHtml("#738A9C");
                    grdResult_Artist.Rows[i].ForeColor =
System.Drawing.ColorTranslator.FromHtml("#F7F7F7");
                    grdResult_Artist.Rows[i].Font.Bold = true;
                }
            }
        }
    }
}

```

Strukturen för gridview som visar resultat från Fetch (hämtad data för uppdatering) och sökning gällande Inkopstabellen.

```

<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Gridview_grdResult_Inkop.ascx.cs"
Inherits="UserControls_Gridview_grdResult_Inkop" %>

```

```

<!-- GridView som visar hel tabell inkl paging etc: -->

<asp:GridView ID="grdResult_Inkop" runat="server" BackColor="White"
    AllowPaging="true" OnRowCommand="gridArtister_Inkop_RowCommand"
HeaderStyle-CssClass="header"
    AllowSorting="false" PageSize="6"
OnPageIndexChanging="grdResult_Inkop_PageIndexChanging"
    AutoGenerateColumns="false">

    <Columns>

        <asp:TemplateField HeaderText="Mark" ItemStyle-CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartist">
            <HeaderTemplate>

                <TF:DerivCheckBox ID="chkInkval_ALL" runat="server" Checked='<%#
Bind("ValInk") %>' AutoPostBack="true"
                UnText='<%#Eval("Skiv_no") %>'
OnCheckedChanged="chkInkvalALLGrdRes_CheckedChanged"
                    EnableViewState="true" />

            </HeaderTemplate>

            <ItemTemplate>

                <TF:DerivCheckBox ID="chkInkval" runat="server" Checked='<%#
Bind("ValInk") %>' AutoPostBack="true"
                UnText='<%#Eval("Skiv_no") %>'
OnCheckedChanged="chkInkval_CheckedChanged" EnableViewState="true" />

            </ItemTemplate>

        </asp:TemplateField>

        <asp:TemplateField ItemStyle-CssClass="grid_style"
            SortExpression="Artist" HeaderStyle-CssClass="pos_grdresult">

            <HeaderTemplate>
                <asp:LinkButton ID="lnkbtnArtist" runat="server"
CommandName="Sort"
                ForeColor="#F7F7F7" CommandArgument="Artist"
OnCommand="SortInkop_Command">
                Artist</asp:LinkButton>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:Button ID="Artist" runat="server" CssClass="arti"
Text='<%#Eval("Artist") %>'
                CommandArgument='<%#Eval("Art_no") %>' CommandName="Artist"
                OnCommand="SortInkop_Command" />
            </ItemTemplate>
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

```

```

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
    <HeaderTemplate>
        <asp:LinkButton ID="lnkbtnAlbum" runat="server" CommandName="Sort"
        ForeColor="#F7F7F7" CommandArgument="Titel"
OnCommand="SortInkop_Command">
            Titel</asp:LinkButton>
    </HeaderTemplate>
    <ItemTemplate>
        <asp:Label ID="lblSortAlbum" runat="server" Text='<%#
Eval("Titel") %>'
        EnableTheming="false">
    </asp:Label>
    </ItemTemplate>
    <ItemStyle CssClass="grid_style" />
</asp:TemplateField>

```

```

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
    <HeaderTemplate>
        <asp:LinkButton ID="lnkbtnForm" runat="server" CommandName="Sort"
        ForeColor="#F7F7F7" CommandArgument="Form"
OnCommand="SortInkop_Command">
            Format</asp:LinkButton>
    </HeaderTemplate>
    <ItemTemplate>
        <asp:Label ID="lblSortForm" runat="server" Text='<%# Eval("Form")
%>'
        EnableTheming="false">
    </asp:Label>
    </ItemTemplate>
    <ItemStyle CssClass="grid_style" />
</asp:TemplateField>

```

```

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
    <HeaderTemplate>
        <asp:LinkButton ID="lnkbtnLand" runat="server" CommandName="Sort"
        ForeColor="#F7F7F7" CommandArgument="Land"
OnCommand="SortInkop_Command">
            Press</asp:LinkButton>
    </HeaderTemplate>
    <ItemTemplate>
        <asp:Label ID="lblSortLand" runat="server" Text='<%# Eval("Land")
%>'
        EnableTheming="false">
    </asp:Label>
    </ItemTemplate>
    <ItemStyle CssClass="grid_style" />
</asp:TemplateField>

```

```

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
CssClass="pos_grdresult">
    <HeaderTemplate>
        <asp:LinkButton ID="lnkbtnUtg" runat="server" CommandName="Sort"

```



```

        ForeColor="#F7F7F7" CommandArgument="Utg"
OnCommand="SortInkop_Command">
        År</asp:LinkButton>
</HeaderTemplate>
<ItemTemplate>
        <asp:Label ID="lblSortUtg" runat="server" Text='<%# Eval("Utg") %>'
        EnableTheming="false">
        </asp:Label>
</ItemTemplate>
        <ItemStyle CssClass="grid_style" />
</asp:TemplateField>

<asp:TemplateField HeaderText="Komtr" HeaderStyle-CssClass="cmt_align"
        ItemStyle-CssClass="grid_style">
        <ItemTemplate>
                <asp:Button ID="kommentar" runat="server" CssClass="cmt"
Text='<%#Eval("Skiv_no") %>'
                CommandArgument='<%#Eval("Skiv_no") %>' CommandName="Kommentar" />
        </ItemTemplate>
        <HeaderStyle CssClass="cmt_align" />
        <ItemStyle CssClass="grid_style" />
</asp:TemplateField>

<asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
        CssClass="pos_grdresult">
        <HeaderTemplate>
                <asp:LinkButton ID="lnkbtnInm_dat" runat="server"
CommandName="Sort"
                ForeColor="#F7F7F7" CommandArgument="Inm_dat"
OnCommand="SortInkop_Command">
                InmDat</asp:LinkButton>
        </HeaderTemplate>
        <ItemTemplate>
                <asp:Label ID="lblSortInm_dat" runat="server" Text='<%#
Eval("Inm_dat", "{0:yyyy-MM-dd}") %>'
                EnableTheming="false" >
                </asp:Label>
        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
</asp:TemplateField>

<asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
        CssClass="pos_grdresult">
        <HeaderTemplate>
                <asp:LinkButton ID="lnkbtnKop_grad" runat="server"
CommandName="Sort"
                ForeColor="#F7F7F7" CommandArgument="Kop_grad"
OnCommand="SortInkop_Command">
                Kgrad</asp:LinkButton>
        </HeaderTemplate>
        <ItemTemplate>
                <asp:Label ID="lblSortKop_grad" runat="server" Text='<%#
Eval("Kop_grad") %>'
                EnableTheming="false" >
                </asp:Label>

```

```

        </ItemTemplate>
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
    CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnKop_kat" runat="server"
    CommandName="Sort"
            ForeColor="#F7F7F7" CommandArgument="Kop_kat"
    OnCommand="SortInkop_Command">
                Kkat</asp:LinkButton>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:Label ID="lblSortKop_kat" runat="server" Text='<%#
    Eval("Kop_kat") %>'
                EnableTheming="false" >
                </asp:Label>
            </ItemTemplate>
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

    <asp:TemplateField ItemStyle-CssClass="grid_style" HeaderStyle-
    CssClass="pos_grdresult">
        <HeaderTemplate>
            <asp:LinkButton ID="lnkbtnCa_pris" runat="server"
    CommandName="Sort"
            ForeColor="#F7F7F7" CommandArgument="Ca_pris"
    OnCommand="SortInkop_Command">
                Capris</asp:LinkButton>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:Label ID="lblSortCa_pris" runat="server" Text='<%#
    Eval("Ca_pris") %>'
                EnableTheming="false" >
                </asp:Label>
            </ItemTemplate>
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

    </Columns>

</asp:GridView>

```

Bakomliggande kod till strukturen för gridview som visar resultat från Fetch (hämtad data för uppdatering) och sökning gällande Inkopstabellen. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Collections;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;
using DB_Callcode_Inkop.Inkop;

public partial class UserControls_Gridview_grdResult_Inkop :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear_Inkop clr = new Clear_Inkop();

    protected void Page_Load(object sender, EventArgs e) { }

    //Kommentarknapp och Artistknapp:
    protected void gridArtister_Inkop_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        //Leta upp kontroller:
        UserControl grd_Artistdata_Inkop_UCtrl =
(UserControl)Page.FindControl("grd_Artistdata_Inkop_UCtrl");

        GridView grd_Artistdata_Inkop =
(GridView)grd_Artistdata_Inkop_UCtrl.FindControl("grd_Artistdata_Inkop");

        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");
        HtmlGenericControl display2 =
(HtmlGenericControl)Page.FindControl("display2");

        try
        {
            dbtn_KomArt kart = new dbtn_KomArt();
            kart.gridArtister_Inkop_RowCommand(grd_Artistdata_Inkop,
display, display2, this.Page, e);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);
        }
    }
}
```

```

        string mess =
"<h2>[Uctrl]UserCgrdResult_Inkop.dbtn_KomArt:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[Uctrl]UserCgrdResult_Inkop.dbtn_KomArt:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        String mess =
"<h2>[Uctrl]UserCgrdResult_Inkop.dbtn_KomArt:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Kryssruta för gridview gridArtister_Inkop och grdResult_Inkop:
protected void chkInkval_CheckedChanged(object sender, EventArgs e)
{
    //Leta upp kontroller:
    UserControl UGridArtister_Inkop_Uctrl =
(UserControl)Page.FindControl("UGridArtister_Inkop_Uctrl");

    GridView gridArtister_Inkop =
(GridView)UGridArtister_Inkop_Uctrl.FindControl("gridArtister_Inkop");

    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        chk_Gridviews cgrd = new chk_Gridviews();
        cgrd.chkInkval_CheckedChanged(gridArtister_Inkop,
grdResult_Inkop);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[Uctrl]UserCgrdResult_Inkop.chk_Gridviews:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {

```

```

        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult_Inkop.chk_Gridviews:MySQLException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult_Inkop.chk_Gridviews:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Headerkryssruta för gridview grdResult_Inkop:
protected void chkInkvalALLGrdRes_CheckedChanged(object sender,
EventArgs e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        hchk_grdResult hchk = new hchk_grdResult();
        hchk.chkInkvalALLGrdRes_CheckedChanged(grdResult_Inkop,
sender);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult_Inkop.hchk_grdResult:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySQLException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCgrdResult_Inkop.hchk_grdResult:MySQLException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);
    }
}

```

```

        String mess =
"<h2>[UCtrl]UserCgrdResult_Inkop.hchk_grdResult:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Sortering för gridview som är programatiskt ansluten till databas
protected void SortInkop_Command(object sender, CommandEventArgs e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    DataSet ds = new DataSet();
    DataView dv = new DataView();

    //Hämta värden så cachekeys kan bli likadana som de vid sökning,
    //dvs individuella cachekeys
    TextBox art_no = (TextBox)Page.FindControl("text3_Artist");
    TextBox album = (TextBox)Page.FindControl("text4_Album");
    DropDownList format =
(DropDownList)Page.FindControl("text5_Format");
    DropDownList press = (DropDownList)Page.FindControl("text6_Press");
    DropDownList ar = (DropDownList)Page.FindControl("text8_ar");
    TextBox kommentar = (TextBox)Page.FindControl("text7_Kommentar");
    TextBox inm_dat = (TextBox)Page.FindControl("text15_Inm_dat");
    DropDownList kop_grad =
(DropDownList)Page.FindControl("text16_Kop_grad");
    DropDownList kop_kat =
(DropDownList)Page.FindControl("text17_Kop_kat");
    TextBox ca_pris = (TextBox)Page.FindControl("text18_Ca_pris");

    ds = (DataSet)HttpRuntime.Cache.Get("cache" + art_no.Text +
album.Text + format.Text + press.Text
        + ar.Text + kommentar.Text + inm_dat.Text + kop_grad.Text +
kop_kat.Text + ca_pris.Text);

    //Första tabellen i datasetet tilldelas till en dataview:
    if (this.Cache["cache" + art_no.Text + album.Text + format.Text +
press.Text
        + ar.Text + kommentar.Text + inm_dat.Text + kop_grad.Text +
kop_kat.Text + ca_pris.Text] == null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        display.InnerHtml = "<h2>Cacheminnets delaytid har gått ut.
Genomför en ny sökning</h2>";
    }
    else
        dv = ds.Tables[0].DefaultView;
}

```

```

        //Kollar om rad är sorterbar, vilken rad som i så fall är
sorterbar, och om
        //den skall sorteras mha stigande eller fallande sorteringsordning:
        if (e.CommandName.Equals("Sort"))
        {
            if (this.ViewState["SortExp"] == null)
            {
                this.ViewState["SortExp"] = e.CommandArgument.ToString();
                this.ViewState["SortOrder"] = "ASC";
            }
            else
            {
                if (this.ViewState["SortExp"].ToString() ==
e.CommandArgument.ToString())
                {
                    if (this.ViewState["SortOrder"].ToString() == "ASC")
                        this.ViewState["SortOrder"] = "DESC";
                    else
                        this.ViewState["SortOrder"] = "ASC";
                }
                else
                {
                    this.ViewState["SortOrder"] = "ASC";
                    this.ViewState["SortExp"] =
e.CommandArgument.ToString();
                }
            }
        }

        //Om 'SortExp' (aktuell kolumnrad) är tillgänglig så sorteras
//den mha dataview enligt inställningarna ovan:
        if (this.ViewState["SortExp"] != null)
        {
            string errmess = "<h2>Ingen tabell i DataView pga
cachminnets ";
            errmess += "delaytid har gått ut. Genomför en ny
sökning</h2>";

            if (dv.Table == null)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces_Inkop(this.Page);

                display.InnerHtml = errmess;
            }
            else
            {
                dv.Sort = this.ViewState["SortExp"].ToString()
                    + " " +
this.ViewState["SortOrder"].ToString();
            }

            //Skapar nytt objekt och adderar sorterade dataview:n till det:
            GridView gr = new GridView();

            //CurrentPageIndex:
            grdResult_Inkop.DataSource = dv;
            grdResult_Inkop.DataBind();
            grdResult_Inkop.Visible = true;

```

```

    }
}

//Paging för gridview som är programatiskt ansluten till databas
protected void grdResult_Inkop_PageIndexChanging(object sender,
GridViewPageEventArgs e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    DataSet ds = new DataSet();
    DataView dv = new DataView();

    //Hämta värden så cachekeys kan bli likadana som de vid sökning,
    //dvs individuella cachekeys
    TextBox art_no = (TextBox)Page.FindControl("text3_Artist");
    TextBox album = (TextBox)Page.FindControl("text4_Album");
    DropDownList format =
(DropDownList)Page.FindControl("text5_Format");
    DropDownList press = (DropDownList)Page.FindControl("text6_Press");
    DropDownList ar = (DropDownList)Page.FindControl("text8_ar");
    TextBox kommentar = (TextBox)Page.FindControl("text7_Kommentar");
    TextBox inm_dat = (TextBox)Page.FindControl("text15_Inm_dat");
    DropDownList kop_grad =
(DropDownList)Page.FindControl("text16_Kop_grad");
    DropDownList kop_kat =
(DropDownList)Page.FindControl("text17_Kop_kat");
    TextBox ca_pris = (TextBox)Page.FindControl("text18_Ca_pris");

    ds = (DataSet)HttpRuntime.Cache.Get("cache" + art_no.Text +
album.Text + format.Text + press.Text
    + ar.Text + kommentar.Text + inm_dat.Text + kop_grad.Text +
kop_kat.Text + ca_pris.Text);

    //Första tabellen i datasetet tilldelas till en dataview:
    if (this.Cache["cache" + art_no.Text + album.Text + format.Text +
press.Text
    + ar.Text + kommentar.Text + inm_dat.Text + kop_grad.Text +
kop_kat.Text + ca_pris.Text] == null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        display.InnerHtml = "<h2>Cacheminnets delaytid har gått ut.
Genomför en ny sökning</h2>";
    }
    else
        dv = ds.Tables[0].DefaultView;
}

```



```

//Om 'SortExp' (aktuell kolumnrad) är tillgänglig så sorteras
//den mha dataview enligt värdena i viewstatevariablerna:
if (this.ViewState["SortExp"] != null)
{
    string errmess = "<h2>Ingen tabell i DataView pga cachminnets
";
    errmess += "delaytid har gått ut. Genomför en ny sökning</h2>";

    if (dv.Table == null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        display.InnerHtml = errmess;
    }
    else
        dv.Sort = this.ViewState["SortExp"].ToString()
            + " " + this.ViewState["SortOrder"].ToString();
}

//Skapar nytt objekt och ett nytt sidindex så man kan bläddra
(paging)
//inklusive state-värden för kryssrutor (RememberOldValues och
RePopulateValues);
GridView gr = new GridView();

gr = (GridView)sender;
gr = grdResult_Inkop;

RememberOldValues();

gr.PageIndex = e.NewPageIndex;

//CurrentPageIndex
grdResult_Inkop.DataSource = dv;
grdResult_Inkop.DataBind();

RePopulateValues();

grdResult_Inkop.Visible = true;

}

```

```

//Sparar kryssrutors state-värden mha bool och Session-objekt:
protected void RememberOldValues()
{
    //Lika många utrymmen för kryssrytevärden som antal rader/sid:
    bool[] values = new bool[grdResult_Inkop.PageSize];
    DerivCheckBox chb;

    //Hittar kryssruta och sparar dess statevärde i array:
    for (int i = 0; i < grdResult_Inkop.Rows.Count; i++)
    {

```

```

        chb =
(DerivCheckBox)grdResult_Inkop.Rows[i].FindControl("chkInkval");
        if (chb != null)
        {
            values[i] = chb.Checked;
        }
    }
    //Sparar bool-array i Session för senare access:
    Session["page" + grdResult_Inkop.PageIndex] = values;
}

//Återskapar bool-array från Sessionobjekt och dess statevärden
//för kryssrutorna på aktuell sida:
protected void RePopulateValues()
{
    if (Session["page" + grdResult_Inkop.PageIndex] != null)
    {
        DerivCheckBox chb;
        bool[] values = (bool[])Session["page" +
grdResult_Inkop.PageIndex];
        for (int i = 0; i < grdResult_Inkop.Rows.Count; i++)
        {
            chb =
(DerivCheckBox)grdResult_Inkop.Rows[i].FindControl("chkInkval");
            chb.Checked = values[i];
            if (chb.Checked == true)
            {
                grdResult_Inkop.Rows[i].BackColor =
System.Drawing.ColorTranslator.FromHtml("#738A9C");
                grdResult_Inkop.Rows[i].ForeColor =
System.Drawing.ColorTranslator.FromHtml("#F7F7F7");
                grdResult_Inkop.Rows[i].Font.Bold = true;
            }
        }
    }
}
}

```

Strukturen för gridview som visar hela tabellens innehåll gällande Skivtabellen.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Gridview_gridArtister.ascx.cs"
Inherits="UserControls_Gridview_gridArtister" %>

<!-- GridView som visar hel tabell inkl paging etc: -->

<asp:GridView ID="gridArtister" runat="server" DataSourceID="main"
    AllowPaging="True" OnRowCommand="gridArtister_RowCommand" HeaderStyle-
CssClass="header"
    AllowSorting="True" PageSize="6" >

    <Columns>

        <asp:TemplateField HeaderText="Mark" ItemStyle-CssClass="grid_style"
            HeaderStyle-CssClass="pos_gridartist">
            <HeaderTemplate>

                <TF:DerivCheckBox ID="chkCDval_ALL" runat="server" Checked='<%#
Bind("Val") %>' AutoPostBack="true"
                UnText='<%#Eval("#") %>'
                OnCheckedChanged="chkCDvalALLMain_CheckedChanged" />

            </HeaderTemplate>

            <ItemTemplate>

                <TF:DerivCheckBox ID="chkCDval" runat="server" Checked='<%#
Bind("Val") %>' AutoPostBack="true"
                UnText='<%#Eval("#") %>'
                OnCheckedChanged="chkCDval_CheckedChanged" />

            </ItemTemplate>

        </asp:TemplateField>

        <asp:TemplateField HeaderText="Artist" ItemStyle-CssClass="grid_style"
            SortExpression="Artist" HeaderStyle-CssClass="pos_gridartist">
            <ItemTemplate>
                <asp:Button ID="Artist" runat="server" CssClass="arti"
Text='<%#Eval("Artist") %>'
                CommandArgument='<%#Eval("Artist_no") %>' CommandName="Artist" />
            </ItemTemplate>
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

        <asp:BoundField DataField="Album" HeaderText="Titel"
            ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
            SortExpression="Album">
            <ItemStyle CssClass="grid_style" />
        </asp:BoundField>
```

```

        <asp:BoundField DataField="Format" HeaderText="Format"
        ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
        SortExpression="Format">
            <ItemStyle CssClass="grid_style" />
        </asp:BoundField>
        <asp:BoundField DataField="Press" HeaderText="Press"
        ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
        SortExpression="Press">
            <ItemStyle CssClass="grid_style" />
        </asp:BoundField>
        <asp:BoundField DataField="Ar" HeaderText="År" HeaderStyle-
        CssClass="pos_gridartist"
        ItemStyle-CssClass="grid_style" SortExpression="Ar">
            <ItemStyle CssClass="grid_style" />
        </asp:BoundField>

        <asp:TemplateField HeaderText="Komtr" HeaderStyle-CssClass="cmt_align"
        ItemStyle-CssClass="grid_style">
            <ItemTemplate>
                <asp:Button ID="kommentar" runat="server" CssClass="cmt"
                Text='<%#Eval("#") %>'
                CommandArgument='<%#Eval("#") %>' CommandName="Kommentar" />
            </ItemTemplate>
            <HeaderStyle CssClass="cmt_align" />
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

    </Columns>

</asp:GridView>

<!-- Obj-datasrc för gridview gridArtister: -->

<asp:ObjectDataSource ID="main" runat="server"
    TypeName="DB_proc.Proc_act" SelectMethod="GetWholeDB"
    EnableCaching="false" CacheDuration="3600"
    OnSelected="ObjectDataSourceMain_Selected">
</asp:ObjectDataSource>

```

Bakomliggande kod till Strukturen för gridview som visar hela tabellens innehåll gällande Skivtabellen. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;
using DB_Callcode.Skivor;

public partial class UserControls_Gridview_gridArtister :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear clr = new Clear();

    protected void Page_Load(object sender, EventArgs e){}

    //Kommentarknapp och Artistknapp:
    protected void gridArtister_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        //Leta upp kontroller:
        UserControl
grd_Artistdata_UCtrl=(UserControl)Page.FindControl("grd_Artistdata_UCtrl");

        GridView grd_Artistdata =
(Gridview)grd_Artistdata_UCtrl.FindControl("grd_Artistdata");

        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");
        HtmlGenericControl display2 =
(HtmlGenericControl)Page.FindControl("display2");

        try
        {
            dbtn_KomArt kart = new dbtn_KomArt();
            kart.gridArtister_RowCommand(grd_Artistdata, display, display2,
this.Page, e);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            string mess =
"<h2>[UCtrl]UserCgridArtister.dbtn_KomArt:FormatException</h2>";

```

```

        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister.dbtn_KomArt:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        String mess =
"<h2>[UCtrl]UserCgridArtister.dbtn_KomArt:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Headerkryssruta för gridview gridArtister:
protected void chkCDvalALLMain_CheckedChanged(object sender, EventArgs
e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        hchk_gridArtister hchk = new hchk_gridArtister();
        hchk.chkCDvalALLMain_CheckedChanged(gridArtister, sender);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister.hchk_gridArtister:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister.hchk_gridArtister:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

```

        catch (System.Exception err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            String mess =
"<h2>[UCtrl]UserCgridArtister.hchk_gridArtister:System.Exception</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
    }

    //Kryssruta för gridview gridArtister och grdResult:
    protected void chkCDval_CheckedChanged(object sender, EventArgs e)
    {
        //Leta upp kontroller:
        UserControl grdResult_UCtrl =
(UserControl)Page.FindControl("grdResult_UCtrl");

        GridView grdResult =
(GridView)grdResult_UCtrl.FindControl("grdResult");

        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

        try
        {
            chk_Gridviews cgrd = new chk_Gridviews();
            cgrd.chkCDval_CheckedChanged(gridArtister, grdResult);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            string mess =
"<h2>[UCtrl]UserCgridArtister.chk_Gridviews:FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            string mess =
"<h2>[UCtrl]UserCgridArtister.chk_Gridviews:MySqlException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (System.Exception err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            String mess =
"<h2>[UCtrl]UserCgridArtister.chk_Gridviews:System.Exception</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
    }

```

```

    }
}

//Fångar exceptions från objectdatasources selectmetod i komponent:
protected void ObjectDataSourceMain_Selected(object sender,
ObjectDataSourceStatusEventArgs e)
{
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    if (e.Exception != null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        gridArtister.Visible = true;

        string mess = "<h2>[UCtrl]UserCgridArtister.ObjektDataSource:
Exception</h2>";
        mess += "<h2>Felmeddelande: <br />" + e.Exception.ToString() +
"</h2>";
        display.InnerHtml = mess;

        e.ExceptionHandled = true;
    }
}

}

```


Strukturen för gridview som visar hela tabellens innehåll gällande Artisttabellen.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Gridview_gridArtister_Artist.ascx.cs"
Inherits="UserControls_Gridview_gridArtister_Artist" %>

<!-- GridView som visar hel tabell inkl paging etc: -->

<asp:GridView ID="gridArtister_Artist" runat="server"
DataSourceID="main_Artist"
    AllowPaging="True" HeaderStyle-CssClass="header"
    AllowSorting="True" PageSize="8"
OnRowCommand="gridArtister_Artist_RowCommand">

    <RowStyle Height="35px" />

    <Columns>

        <asp:TemplateField HeaderText="Mark" ItemStyle-CssClass="grid_style"
HeaderStyle-CssClass="pos_gridartist">
            <HeaderTemplate>

                <TF:DerivCheckBox ID="chkArtval_ALL" runat="server" Checked="<%#
Bind("ValArt") %>" AutoPostBack="true"
                UnText="<%#Eval("No") %>"
OnCheckedChanged="chkArtvalALLMain_CheckedChanged" />

            </HeaderTemplate>

            <ItemTemplate>

                <TF:DerivCheckBox ID="chkArtval" runat="server" Checked="<%#
Bind("ValArt") %>" AutoPostBack="true"
                UnText="<%#Eval("No") %>"
OnCheckedChanged="chkArtval_CheckedChanged" />

            </ItemTemplate>

        </asp:TemplateField>

        <asp:BoundField DataField="No" HeaderText="ArtNo"
            ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
            SortExpression="No">
            <ItemStyle CssClass="grid_style" />
        </asp:BoundField>

        <asp:TemplateField HeaderText="Artist" ItemStyle-CssClass="grid_style"
SortExpression="Artist" HeaderStyle-CssClass="pos_gridartist">
            <ItemTemplate>
                <asp:Button ID="Artist" runat="server" CssClass="arti"
Text="<%#Eval("Artist") %>"
                CommandArgument="<%#Eval("Artist") %>" CommandName="Artist_kortbio" />
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
```

```

        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

    <asp:BoundField DataField="VPfr" HeaderText="VPfr "
        ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
        SortExpression="VPfr">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="VPtill" HeaderText="VPtill"
        ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
        SortExpression="VPtill">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="Ursland" HeaderText="Ursland"
        ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
        SortExpression="Ursland">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="Ursstad" HeaderText="Ursstad" HeaderStyle-
        CssClass="pos_gridartist"
        ItemStyle-CssClass="grid_style" SortExpression="Ursstad">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="Musiktyp" HeaderText="Musiktyp" HeaderStyle-
        CssClass="pos_gridartist"
        ItemStyle-CssClass="grid_style" SortExpression="Musiktyp">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>

</Columns>

</asp:GridView>

<!-- Obj-datasrc för gridview gridArtister: -->
<asp:ObjectDataSource ID="main_Artist" runat="server"
    TypeName="DB_proc_Artist.Artist" SelectMethod="GetWholeDB"
    EnableCaching="false" CacheDuration="3600"
    OnSelected="ObjectDataSourceMain_Selected">
</asp:ObjectDataSource>

```

Bakomliggande kod till Strukturen för gridview som visar hela tabellens innehåll gällande Artisttabellen. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

using DB_Callcode_Artist.Artist;

public partial class UserControls_Gridview_gridArtister_Artist :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear_Artist clr = new Clear_Artist();

    protected void Page_Load(object sender, EventArgs e){}

    //Artistknapp:
    protected void gridArtister_Artist_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        //Leta upp kontroller:

        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

        try
        {
            dbtn_Visa_Kortbio vkb = new dbtn_Visa_Kortbio();
            vkb.grd_Artistdata_Artist_RowCommand(display, e, this.Page);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);

            string mess =
"<h2>[UCtrl]UserCgridArtister_Artist.dbtn_KomArt:FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
        }
    }
}
```

```

        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister_Artist.dbtn_KomArt:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        String mess =
"<h2>[UCtrl]UserCgridArtister_Artist.dbtn_KomArt:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Headerkryssruta för gridview gridArtister:
protected void chkArtvalALLMain_CheckedChanged(object sender, EventArgs
e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        hchk_gridArtister hchk = new hchk_gridArtister();
        hchk.chkArtvalALLMain_CheckedChanged(gridArtister_Artist,
sender);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister_Artist.hchk_gridArtister:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister_Artist.hchk_gridArtister:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews

```

```

        clr.Clean_surfaces_Artist(this.Page);

        String mess =
"<h2>[Uctrl]UserCgridArtister_Artist.hchk_gridArtister:System.Exception</h2>"
">";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }

    //Kryssruta för gridview gridArtister och grdResult:
    protected void chkArtval_CheckedChanged(object sender, EventArgs e)
    {
        //Leta upp kontroller:
        UserControl grdResult_Artist_UCtrl =
(UserControl)Page.FindControl("grdResult_Artist_UCtrl");

        GridView grdResult_Artist =
(GridView)grdResult_Artist_UCtrl.FindControl("grdResult_Artist");

        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

        try
        {
            chk_Gridviews cgrd = new chk_Gridviews();
            cgrd.chkArtval_CheckedChanged(gridArtister_Artist,
grdResult_Artist);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);

            string mess =
"<h2>[Uctrl]UserCgridArtister_Artist.chk_Gridviews:FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);

            string mess =
"<h2>[Uctrl]UserCgridArtister_Artist.chk_Gridviews:MySqlException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (System.Exception err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);

            String mess =
"<h2>[Uctrl]UserCgridArtister_Artist.chk_Gridviews:System.Exception</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
    }

```

```

    }

    //Fångar exceptions från objectdatasources selectmetod i komponent:
    protected void ObjectDataSourceMain_Selected(object sender,
ObjectDataSourceStatusEventArgs e)
    {
        //Leta upp kontroller:
        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

        if (e.Exception != null)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);

            gridArtister_Artist.Visible = true;

            string mess =
"<h2>[UCtrl]UserCgridArtister_Artist.ObjektDataSource: Exception</h2>";
            mess += "<h2>Felmeddelande: <br />" + e.Exception.ToString() +
"</h2>";
            display.InnerHtml = mess;

            e.ExceptionHandled = true;
        }
    }

}

```

Strukturen för gridview som visar hela tabellens innehåll gällande Inkopstabellen.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Gridview_gridArtister_Inkop.ascx.cs"
Inherits="UserControls_Gridview_gridArtister_Inkop" %>

<!-- GridView som visar hel tabell inkl paging etc: -->

<asp:GridView ID="gridArtister_Inkop" runat="server"
DataSourceID="main_Inkop"
AllowPaging="True" OnRowCommand="gridArtister_Inkop_RowCommand"
HeaderStyle-CssClass="header"
AllowSorting="True" PageSize="6" >

    <Columns>

        <asp:TemplateField HeaderText="Mark" ItemStyle-CssClass="grid_style"
HeaderStyle-CssClass="pos_gridartist">
            <HeaderTemplate>

                <TF:DerivCheckBox ID="chkInkval_ALL" runat="server"
Checked='<%# Bind("ValInk") %>' AutoPostBack="true"
                UnText='<%#Eval("Skiv_no") %>'
OnCheckedChanged="chkInkvalALLMain_CheckedChanged" />

            </HeaderTemplate>

            <ItemTemplate>

                <TF:DerivCheckBox ID="chkInkval" runat="server" Checked='<%#
Bind("ValInk") %>' AutoPostBack="true"
                UnText='<%#Eval("Skiv_no") %>'
OnCheckedChanged="chkInkval_CheckedChanged" />

            </ItemTemplate>

        </asp:TemplateField>

        <asp:TemplateField HeaderText="Artist" ItemStyle-CssClass="grid_style"
SortExpression="Artist" HeaderStyle-CssClass="pos_gridartist">
            <ItemTemplate>
                <asp:Button ID="Artist" runat="server" CssClass="arti"
Text='<%#Eval("Artist") %>'
                CommandArgument='<%#Eval("Art_no") %>' CommandName="Artist" />
            </ItemTemplate>
            <ItemStyle CssClass="grid_style" />
        </asp:TemplateField>

        <asp:BoundField DataField="Titel" HeaderText="Titel"
ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
SortExpression="Titel">
```

```

        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="Form" HeaderText="Format"
    ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
    SortExpression="Form">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="Land" HeaderText="Press"
    ItemStyle-CssClass="grid_style" HeaderStyle-CssClass="pos_gridartist"
    SortExpression="Land">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="Utg" HeaderText="År" HeaderStyle-
    CssClass="pos_gridartist"
    ItemStyle-CssClass="grid_style" SortExpression="Utg">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>

    <asp:TemplateField HeaderText="Komtr" HeaderStyle-CssClass="cmt_align"
    ItemStyle-CssClass="grid_style">
        <ItemTemplate>
            <asp:Button ID="kommentar" runat="server" CssClass="cmt"
Text='<%#Eval("Skiv_no") %>'
            CommandArgument='<%#Eval("Skiv_no") %>' CommandName="Kommentar"
        />
        </ItemTemplate>
        <HeaderStyle CssClass="cmt_align" />
        <ItemStyle CssClass="grid_style" />
    </asp:TemplateField>

    <asp:BoundField DataField="Inm_dat" HeaderText="InmDat"
    DataFormatString="{0:yyyy-MM-dd}"
    HtmlEncode="false" HeaderStyle-CssClass="pos_gridartist" ItemStyle-
    CssClass="grid_style"
    SortExpression="Inm_dat">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="Kop_grad" HeaderText="Kgrad" HeaderStyle-
    CssClass="pos_gridartist"
    ItemStyle-CssClass="grid_style" SortExpression="Kop_grad">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="Kop_kat" HeaderText="Kkat" HeaderStyle-
    CssClass="pos_gridartist"
    ItemStyle-CssClass="grid_style" SortExpression="Kop_kat">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>
    <asp:BoundField DataField="Ca_pris" HeaderText="Capris" HeaderStyle-
    CssClass="pos_gridartist"
    ItemStyle-CssClass="grid_style" SortExpression="Ca_pris">
        <ItemStyle CssClass="grid_style" />
    </asp:BoundField>

    </Columns>

</asp:GridView>

```



```
<!-- Obj-datasrc för gridview gridArtister: -->  
  
<asp:ObjectDataSource ID="main_Inkop" runat="server"  
    TypeName="DB_proc_Inkop.Inkop" SelectMethod="GetWholeDB"  
    EnableCaching="false" CacheDuration="3600"  
    OnSelected="ObjectDataSourceMain_Selected">  
</asp:ObjectDataSource>
```

Bakomliggande kod till Strukturen för gridview som visar hela tabellens innehåll gällande Inkopstabellen. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

using DB_Callcode_Inkop.Inkop;

public partial class UserControls_Gridview_gridArtister_Inkop :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear_Inkop clr = new Clear_Inkop();

    protected void Page_Load(object sender, EventArgs e) { }

    //Kommentarknapp och Artistknapp:
    protected void gridArtister_Inkop_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        //Leta upp kontroller:
        UserControl grd_Artistdata_Inkop_UCtrl =
(UserControl)Page.FindControl("grd_Artistdata_Inkop_UCtrl");

        GridView grd_Artistdata_Inkop =
(GridView)grd_Artistdata_Inkop_UCtrl.FindControl("grd_Artistdata_Inkop");

        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");
        HtmlGenericControl display2 =
(HtmlGenericControl)Page.FindControl("display2");

        try
        {
            dbtn_KomArt kart = new dbtn_KomArt();
            kart.gridArtister_Inkop_RowCommand(grd_Artistdata_Inkop,
display, display2, this.Page, e);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);
        }
    }
}
```

```

        string mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.dbtn_KomArt:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.dbtn_KomArt:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        String mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.dbtn_KomArt:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Headerkryssruta för gridview gridArtister_Inkop:
protected void chkInkvalALLMain_CheckedChanged(object sender, EventArgs
e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        hchk_gridArtister hchk = new hchk_gridArtister();
        hchk.chkInkvalALLMain_CheckedChanged(gridArtister_Inkop,
sender);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.hchk_gridArtister:FormatException</h2>"
;
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

```

```

        string mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.hchk_gridArtister:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        String mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.hchk_gridArtister:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Kryssruta för gridview gridArtister_Inkop och grdResult_Inkop:
protected void chkInkval_CheckedChanged(object sender, EventArgs e)
{
    //Leta upp kontroller:
    UserControl grdResult_Inkop_UCtrl =
(UserControl)Page.FindControl("grdResult_Inkop_UCtrl");

    GridView grdResult_Inkop =
(GridView)grdResult_Inkop_UCtrl.FindControl("grdResult_Inkop");

    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        chk_Gridviews cgrd = new chk_Gridviews();
        cgrd.chkInkval_CheckedChanged(gridArtister_Inkop,
grdResult_Inkop);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.chk_Gridviews:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.chk_Gridviews:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```

```

    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        String mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.chk_Gridviews: System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Fångar exceptions från objectdatasources selectmetod i komponent:
protected void ObjectDataSourceMain_Selected(object sender,
ObjectDataSourceStatusEventArgs e)
{
    //Leta upp kontrollern:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    if (e.Exception != null)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        gridArtister_Inkop.Visible = true;

        string mess =
"<h2>[UCtrl]UserCgridArtister_Inkop.ObjektDataSource: Exception</h2>";
        mess += "<h2>Felmeddelande: <br />" + e.Exception.ToString() +
"</h2>";
        display.InnerHtml = mess;

        e.ExceptionHandled = true;
    }
}

}

```

Strukturen för panel som i Skivtabellen möjliggör smidigare hantering av längre kommentarer beträffande skivor.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Panel_Kommentar.ascx.cs" Inherits="UserControls_Panel_Kommentar"
%>

<!-- Administrering av längre kommentarer: -->

<asp:Panel ID="pnlKommentar" runat="server" Visible="false">

    <asp:TextBox ID="txtKommentar" TextMode="MultiLine"
MaxLength="6000"
    runat="server" CssClass="txtKortbio">
    </asp:TextBox>

    <asp:Button ID="dynbtnKommentar" runat="server" CssClass="cmt
dynbtnKortbio"
    Text="Till" OnClick="dynbtnKommentar_OnClick" />

    <asp:Button ID="dynbtnFetchKommentar" runat="server" CssClass="cmt
dynbtnFetchKbio"
    Text="Från" OnClick="dynbtnFetchKommentar_OnClick" />

    <asp:Button ID="dynbtnRensaSkarm_Kom" runat="server" CssClass="cmt
dynbtnRensaSkarm"
    Text="Rensa" OnClick="dynbtnRensaSkarm_Kom_OnClick" />

    <asp:Label ID="lblKommentarNo" runat="server" CssClass="label1
lblArtistNo"
    Text="Kommentarno"></asp:Label>

    <asp:TextBox ID="txtKommentarNo" CssClass="txtArtistNo"
runat="server" Size="1">
    </asp:TextBox>

</asp:Panel>
```

Bakomliggande kod till strukturen för panel som i Skivtabellen möjliggör smidigare hantering av längre kommentarer beträffande skivor. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;
using DB_Callcode.Skivor;

public partial class UserControls_Panel_Kommentar :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear clr = new Clear();

    protected void Page_Load(object sender, EventArgs e) { }

    //Matar in längre kommentarer till skivtabell:
    protected void dynbtnKommentar_OnClick(object sender, EventArgs e)
    {
        //Leta upp kontroller:
        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

        try
        {
            dbtn_Kommentar_Input kin = new dbtn_Kommentar_Input();
            kin.dynbtnKommentar(display, txtKommentar, txtKommentarNo,
this.Page);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);

            string mess =
"<h2>[UCtrl]UserCpnlKommentar.dbtn_Kommentar_Input:FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(this.Page);
        }
    }
}
```

```

        string mess =
"<h2>[UCtrl]UserCpnlKommentar.dbtn_Kommentar_Input:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        String mess =
"<h2>[UCtrl]UserCpnlKommentar.dbtn_Kommentar_Input:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Hämtar längre kommentarer från skivtabell (via kommentarpanel):
protected void dynbtnFetchKommentar_OnClick(object sender, EventArgs e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        dbtn_Kommentar_Output kout = new dbtn_Kommentar_Output();
        kout.dynbtnKommentarFetch(display, pnlKommentar, txtKommentar,
txtKommentarNo, this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[UCtrl]UserCpnlKommentar.dbtn_Kommentar_Output:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);

        string mess =
"<h2>[UCtrl]UserCpnlKommentar.dbtn_Kommentar_Output:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(this.Page);
    }
}

```



```

        String mess =
"<h2>[UCtrl]UserCpnlKommentar.dbtn_Kommentar_Output:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;

    }

}

//Rensar textytan och kommentarnummerrutan från text hos
kommentarpanelen:
protected void dynbtnRensaSkarm_Kom_OnClick(object sender, EventArgs e)
{
    txtKommentar.Text = "";
    txtKommentarNo.Text = "";
}
}

```

Strukturen för panel som i Artisttabellen möjliggör smidigare hantering av artisters kortbiografier.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Panel_Kortbio.ascx.cs" Inherits="UserControls_Panel_Kortbio" %>

<!-- Kortbio-multiline-textruta och knapp för inmatning till DB: -->

<asp:Panel ID="pnlKortbio" runat="server" Visible="false">

    <asp:TextBox ID="txtKortbio" TextMode="MultiLine"
MaxLength="6000"
    runat="server" CssClass="txtKortbio">
    </asp:TextBox>

    <asp:Button ID="dynbtnKortbio" runat="server" CssClass="cmt
dynbtnKortbio"
    Text="Till" OnClick="dynbtnKortbio_OnClick" />

    <asp:Button ID="dynbtnFetchKbio" runat="server" CssClass="cmt
dynbtnFetchKbio"
    Text="Från" OnClick="dynbtnKortbioFetch_OnClick" />

    <asp:Button ID="dynbtnRensaSkarm" runat="server" CssClass="cmt
dynbtnRensaSkarm"
    Text="Rensa" OnClick="dynbtnRensaSkarmArtist_OnClick" />

    <asp:Label ID="lblArtistNo" runat="server" CssClass="labell
lblArtistNo"
    Text="Artno"></asp:Label>

    <asp:TextBox ID="txtArtistNo" CssClass="txtArtistNo"
runat="server" Size="1">
    </asp:TextBox>

</asp:Panel>
```

Bakomliggande kod till strukturen för panel som i Artisttabellen möjliggör smidigare hantering av artisters kortbiografier. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

using DB_Callcode_Artist.Artist;

public partial class UserControls_Panel_Kortbio : System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear_Artist clr = new Clear_Artist();

    protected void Page_Load(object sender, EventArgs e) {}

    //Matar in data till kortibo (via kortbiopanel) i artist-tabell:
    protected void dynbtnKortbio_OnClick(object sender, EventArgs e)
    {
        //Leta upp kontrollen:
        HtmlGenericControl display =
        (HtmlGenericControl)Page.FindControl("display");

        try
        {
            dbtn_Kortbio_Input kbin = new dbtn_Kortbio_Input();
            kbin.dynbtnKortbio(display, txtKortbio, txtArtistNo,
this.Page);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);

            string mess =
"<h2>[UCtrl]UserCpnlKortbio_Artist.dbtn_Kortbio_Input:FormatException</h2>"
;
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(this.Page);
        }
    }
}
```

```

        string mess =
"<h2>[UCtrl]UserCpnlKortbio_Artist.dbtn_Kortbio_Input:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        String mess =
"<h2>[UCtrl]UserCpnlKortbio_Artist.dbtn_Kortbio_Input:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Hämtar kortio från artist-tabell (via kortbiopanel) i artist-tabell::
protected void dynbtnKortbioFetch_OnClick(object sender, EventArgs e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        dbtn_Kortbio_Output kbout = new dbtn_Kortbio_Output();
        kbout.dynbtnKortbioFetch(display, pnlKortbio, txtKortbio,
        txtArtistNo, this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[UCtrl]UserCpnlKortbio_Artist.dbtn_Kortbio_Output:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        string mess =
"<h2>[UCtrl]UserCpnlKortbio_Artist.dbtn_Kortbio_Output:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)

```

```

    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(this.Page);

        String mess =
"<h2>[UCtrl]UserCpnlKortbio_Artist.dbtn_Kortbio_Output: System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Rensar textytan och artistnummerrutan från text hos kortbiopanelen:
protected void dynbtnRensaSkarmArtist_OnClick(object sender, EventArgs
e)
{
    txtKortbio.Text = "";
    txtArtistNo.Text = "";
}
}

```

Strukturen för panel som i Inkopstabellen möjliggör smidigare hantering av längre kommentarer beträffande skivor som skall köpas in.

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Panel_Kommentar_Inkop.ascx.cs"
Inherits="UserControls_Panel_Kommentar_Inkop" %>

<!-- Administrering av längre kommentarer: -->

<asp:Panel ID="pnlKommentar_Inkop" runat="server" Visible="false">

    <asp:TextBox ID="txtKommentar" TextMode="MultiLine" MaxLength="6000"
        runat="server" CssClass="txtKortbio">
    </asp:TextBox>

    <asp:Button ID="dynbtnKommentar" runat="server" CssClass="cmt
dynbtnKortbio"
        Text="Till" OnClick="dynbtnKommentar_OnClick" />

    <asp:Button ID="dynbtnFetchKommentar" runat="server" CssClass="cmt
dynbtnFetchKbio"
        Text="Från" OnClick="dynbtnFetchKommentar_OnClick" />

    <asp:Button ID="dynbtnRensaSkarm_Kom" runat="server" CssClass="cmt
dynbtnRensaSkarm"
        Text="Rensa" OnClick="dynbtnRensaSkarm_Kom_OnClick" />

    <asp:Label ID="lblKommentarNo" runat="server" CssClass="label1
lblArtistNo"
        Text="Kommentarno"></asp:Label>

    <asp:TextBox ID="txtKommentarNo" CssClass="txtArtistNo"
runat="server" Size="1">
    </asp:TextBox>

</asp:Panel>
```

Bakomliggande kod till strukturen för panel som i Inkopstabellen möjliggör smidigare hantering av längre kommentarer för skivor som skall köpas in. Här sker anrop till applikationens mellanskikt där funktioner i komponenter tillgodoser dess behov.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;
using DB_Callcode_Inkop.Inkop;

public partial class UserControls_Panel_Kommentar_Inkop :
System.Web.UI.UserControl
{
    //Instans för rensning av display och gridviews
    Clear_Inkop clr = new Clear_Inkop();

    protected void Page_Load(object sender, EventArgs e) { }

    //Matar in längre kommentarer till Inkoptabell:
    protected void dynbtnKommentar_OnClick(object sender, EventArgs e)
    {
        //Leta upp kontroller:
        HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

        try
        {
            dbtn_Kommentar_Input kin = new dbtn_Kommentar_Input();
            kin.dynbtnKommentar(display, txtKommentar, txtKommentarNo,
this.Page);
        }
        catch (FormatException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);

            string mess =
"<h2>[UCtrl]UserCpnlKommentar_Inkop.dbtn_Kommentar_Input:FormatException</h2>";
            mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
            display.InnerHtml = mess;
        }
        catch (MySqlException err)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(this.Page);
        }
    }
}
```

```

        string mess =
"<h2>[UCtrl]UserCpnlKommentar_Inkop.dbtn_Kommentar_Input:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        String mess =
"<h2>[UCtrl]UserCpnlKommentar_Inkop.dbtn_Kommentar_Input:System.Exception</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Hämtar längre kommentarer från Inkoptabell (via kommentarpanel):
protected void dynbtnFetchKommentar_OnClick(object sender, EventArgs e)
{
    //Leta upp kontroller:
    HtmlGenericControl display =
(HtmlGenericControl)Page.FindControl("display");

    try
    {
        dbtn_Kommentar_Output kout = new dbtn_Kommentar_Output();
        kout.dynbtnKommentarFetch( display, pnlKommentar_Inkop,
txtKommentar,
        txtKommentarNo, this.Page);
    }
    catch (FormatException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCpnlKommentar_Inkop.dbtn_Kommentar_Output:FormatException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
    catch (MySqlException err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        string mess =
"<h2>[UCtrl]UserCpnlKommentar_Inkop.dbtn_Kommentar_Output:MySqlException</h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

```



```

    }
    catch (System.Exception err)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(this.Page);

        String mess =
"<h2>[UCtrl]UserCpnlKommentar_Inkop.dbtn_Kommentar_Output:System.Exception<
/h2>";
        mess += "<h2>Felmeddelande: " + err.Message + "</h2>";
        display.InnerHtml = mess;
    }
}

//Rensar textytan och kommentarnummerrutan från text hos
kommentarpanelen:
protected void dynbtnRensaSkarm_Kom_OnClick(object sender, EventArgs e)
{
    txtKommentar.Text = "";
    txtKommentarNo.Text = "";
}

}

```

Frontend - Themes

Hanterar de inställningar för förekommande gridviews i applikationen som är gemensamma, samt samma hantering för labels.

Theme som hanterar de inställningar för förekommande gridviews i applikationen som är gemensamma, samt samma hantering för labels.

```
<asp:GridView runat="server" BackColor="White"
    BorderColor="#E7E7FF" BorderStyle="None" BorderWidth="1px"
    GridLines="Horizontal" Visible="false"
    AutoGenerateColumns="false" Font-Size="1em">

    <HeaderStyle BackColor="#990033" Font-Bold="True" Font-Size="1.45em"
    ForeColor="#F7F7F7"
        Font-Names="Garamond, Times New Roman, Georgia, Serif" />
    <RowStyle BackColor="#F7F7F7" ForeColor="#4A3C8C" />
    <FooterStyle BackColor="#B5C7DE" ForeColor="#4A3C8C" />
    <PagerStyle BackColor="#990033" ForeColor="#F7F7F7"
    HorizontalAlign="Right" Font-Size="1em" />

</asp:GridView>

<asp:label runat="server" Font-Bold="True" Font-Size="1.5em" />
```

Frontend - JavaScript

Serverside-scriptspråk som ger utökad funktionalitet i aktuell webbläsare. I min applikation är de två till antalet.

Javascript

Javascript som känner av vilken av de tre webbläsaren som applikationen är utvecklad för som användaren nyttjar, och adresserar i enlighet med det till motsvarande CSS-fil.

```
<!--
browser_version= parseInt(navigator.appVersion);
browser_type = navigator.appName;
agentheadz_ff=navigator.userAgent.indexOf("Firefox")
agentheadz_cr=navigator.userAgent.indexOf("Chrome")

if (browser_type == "Microsoft Internet Explorer" && (browser_version >=
4))
{
document.write("<link REL='stylesheet' HREF='styles/style_IE.css'
TYPE='text/css'>");
}

else if (browser_type == "Netscape" && (browser_version >= 4)&&
agentheadz_ff!=-1)
{
document.write("<link REL='stylesheet' HREF='styles/style_FF.css'
TYPE='text/css'>");
}

else if (browser_type == "Netscape" && agentheadz_cr!=-1)
{
document.write("<link REL='stylesheet' HREF='styles/style_CR.css'
TYPE='text/css'>");
}

// -->
```

Javascript

Javascript som ger aktuellt datum och tid för tids/datum-visningen intill rubriken för applikationen.

```
<!--

function datum()
{
//Inhämtande av tidsuppgifter:
nydatum = new Date();
var year = nydatum.getFullYear();
var month = nydatum.getMonth()+1;
var day = nydatum.getDate();
var hour=nydatum.getHours();
var minute=nydatum.getMinutes();
var second=nydatum.getSeconds();

//Tidsjusteringar:
year=(year< 1000 ? 1900 : "" )+year;
month=(month< 10 ? "0" : "" )+month;
day=(day< 10 ? "0" : "" )+day;
hour=(hour < 10 ? "0" : "" )+hour;
minute=( minute < 10 ? "0" : "" )+minute;
second=(second< 10 ? "0" : "" )+second;

//Skapande av tids-sträng:
var timestring=year+"-"+month+"-"+day+" "+hour+": "+minute+": "+second;

//Utskrift:
document.getElementById("clock").firstChild.nodeValue = timestring;
}

// -->
```

Frontend - CSS-filer

Cascading Style Sheets (stilmallar) är ett språk som hjälper till utforma grafiska detaljer samt layout med mera. Just för den här applikationen i enlighet med den webbläsare som aktuell CSS-fil är anpassad efter

Mer generellt och exempelvis Kan CSS-filer nyttjas för anpassning efter kriterier som skärmapplösning, typsnitt, textstorlek eller som i mitt fall enligt ovan typ av webbläsare, och så vidare.

CSS

Fil som anpassar grafiska detaljer samt layout med mera i enlighet med den webbläsare som aktuell css-fil är anpassad efter. **Den här är för Google Chrome.**

```
/*LAYERS*/

#container
{
    width:712px;
    height:640px;
    position:absolute;
    left:50%;
    margin-left:-356px;
}

#header
{
    height:40px;
    width:711px;
    background: transparent url(../bakgrundsbilder/header3.gif) no-
repeat center bottom;
    float:left;
    color:#fff;
    font-size:12px;
}

#content
{
    background:#D8D8D8;
    float:left;
    color:#000;
    font-size:12px;
    width:711px;
    height:421px;
}

#navigation
{
    background: transparent url(../bakgrundsbilder/nav_bakgrund3.gif)
no-repeat center bottom;
    color:#fff;
    font-size:12px;
    height:170px;
    width:712px;
    float:left;
}

/*DATUMVISNING*/
#datumruta
{
    position:absolute;
    right:50px;
    top:10px;
}
```



```

    font-weight:bold;
    color:#990033;
    background: #D8D8D8 no-repeat center bottom;
    width:121px;
    height:12px;
    border: 4px solid #fff;
    font-size:1.1em;
    font-family: Times New Roman, Georgia, Garamond, serif;
    padding-left:2px;
    padding-bottom:3px;
}

/*DISPLAY*/

.ruta{
    width:662px;
    height:370px;
    border: 5px solid #000;
    background:#D8D8D8;
    float:left;
    margin-right:0px;
    margin-left:10px;
    margin-right:10px;
    margin-bottom:10px;
    margin-top:10px;
    padding:10px;
    overflow:auto;
    font-size:1em;
    font-family: Times New Roman, Georgia, Garamond, serif;
}

/*TEXTFORMATERING:*/

/*KORTBIO:*/

#visakortbiodata
{
    margin-right:240pt;
    color:#000;
}
/*#bredd{margin-right:280pt;}*/

/*KOMMENTAR:*/
#visakommentar
{
    margin-right:240pt;
    color:#000;
}

```

```

/*KORTBIOPANEL_INPUT:*/
#kortbiopanel_input
{
    margin-right:240pt;
    color:#000;
}

/*KOMMENTARPANEL_INPUT:*/
#kommentarpanel_input
{
    margin-right:240pt;
    color:#000;
}


/*DYNAMISKA BIOGRAFI BILDER*/

.dyn_pic
{
    border: 8px solid #990033;
    display: block;
}


/*HÅLLER NAVRIGERINGSOBJEKTEN (KONTROLLYTAN) PÅ PLATS*/

.move{margin-left:-16.4%;}


/*LABELS-KONTROLLYTA-RAD 1*/

.label1{color:#fff;bottom:152px;position:absolute;}

#label1_Artist {left:150px;}
#label2_Album {left:285px;}
#label3_Format {left:418px;}
#label4_Press {left:545px;}
#label5_Kommentar {left:675px;}
#label6_ar {left:612px;}

```

```

/*TEXTUTOR-KONTROLLYTA-RAD 1*/

.text1{position:absolute;bottom:130px;}

#text3_Artist{left:150px;}
#text4_Album{left:285px;}
#text5_Format{left:418px;}
#text6_Press{left:545px;}
#text7_Kommentar{left:675px;}
#text8_ar{left:612px;}

/*LABELS-KONTROLLYTA-RAD 2*/

.label2{color:#fff;bottom:105px;position:absolute;}

#label7_VPfr {left:150px;}
#label8_VPtill {left:215px;}
#label9_Ursland {left:285px;}
#label10_Ursstad {left:352px;}
#label11_Musiktyp{left:430px;}
#label12_Kortbio{left:525px;}
#label13_Inm_dat{left:603px;}
#label14_Kop_grad{left:680px;}
#label15_Kop_kat{left:720px;}
#label16_Ca_pris{left:775px;}

/*TEXTUTOR-KONTROLLYTA-RAD 2*/

.text2{position:absolute;bottom:85px;}

#text9_VPfr{left:150px;}
#text10_VPtill{left:215px;}

```

```

#text11_Ursland{left:285px;}

#text12_Ursstad{left:352px;}

#text13_Musiktyp{left:430px;}

#text14_Kortbio{left:525px;}

#text15_Inm_dat{left:603px;}

#text16_Kop_grad{left:680px;}

#text17_Kop_kat{left:720px;}

#text18_Ca_pris{left:775px;}


/*KNAPPPAR-KONTROLLYTA*/

.input
{
    font-style: normal;
    font-weight:bold;
    color: #006400;
    font-size:0.8em;
    font-family: Times New Roman, Georgia, Garamond, serif;
    height:29px;
    background: transparent url(../knappar/knapp_nav.gif) no-repeat center
bottom;
    border:0;
    width:80px;
    position:absolute;
}

input.input:hover{color:#990033;}


/*DYNAMISK GEN KOMMENTAR-KNAPP*/

.cmt
{
    background: transparent url(../knappar/tryck_knapp.gif) no-repeat
center bottom;
    border:0;
    width:70px;
    height:33px;
    color:#fff;
    font-weight:bold;
    font-size:0.8em;
    font-family: Times New Roman, Georgia, Garamond, serif;
}

input.cmt:hover{color:#fff; background: transparent
url(../knappar/tryck_knapp_hover.gif) no-repeat center bottom;}


/*DYNAMISK GEN ARTIST-KNAPP*/

```

```

.arti
{
    background: transparent url(../knappar/knapp_artist3.gif) no-repeat
center bottom;
    border:0;
    width:130px;
    height:20px;
    color:#990033;
    font-size:0.8em;
    font-family: Times New Roman, Georgia, Garamond, serif;
}

input.arti:hover{color:#006400; background: transparent
url(../knappar/knapp_artist3_hover.gif) no-repeat center bottom;}

```

```

/*RAD 1*/

.rad1{bottom:52px;}

#button_visa_databas{left:150px;}

#button_lagg_till{left:230px;}

#button_ta_bort{left:310px;}

#button_uppdatera{left:390px;}

#button_sok{left:470px;}

```

```

/*RAD 2*/

.rad2{bottom:27px;}

#btn21{left:150px;}

#btn22{left:230px;}

#btn23{left:310px;}

#btn24{left:390px;}

#btn25{left:470px;}

```

```

/*GENERELL FORMATERING*/

body
{
    background-color:#863E58;
    background-image:url(../bakgrundsbilder/bg_toning2.gif);
    background-repeat: repeat-x;

    /* Antal varianter på text-layout: */
    /*font-family: verdana, arial, sans-serif;*/
    /*font-family: Times New Roman, verdana, sans-serif;*/
    /*font-family: Times New Roman, Georgia, Garamond, serif;*/
    font-family:Garamond, Times New Roman, Georgia, serif;

}

a:link {color:#000; text-decoration: underline; }
a:active {color:#000; text-decoration: underline; }
a:visited {color:#000; text-decoration: underline; }
a:hover {color:#990033; text-decoration: underline; }

H1 {font-size:16pt;color:#fff;background-color:#990033;}
H2 {font-size:13pt; color:#000;}
H3 {font-size:10pt; color:#006400;}


/* VALKNAPPAR (Radio-buttons): */

.optSkivtabell
{
    position:absolute;
    bottom:60px;
    left:460px;
}

.optArtisttabell
{
    position:absolute;
    bottom:40px;
    left:460px;
}

.optInkopstabell
{
    position:absolute;
    bottom:20px;
    left:460px;
}


/* LABELS TILL VALKNAPPAR (Radio-buttons): */

.label3{color:#fff;position:absolute;font-weight:bold;font-size:1em;}

```

```

#label13_TBskivor{left:600px;bottom:60px;}
#label14_TBartist{left:600px;bottom:40px;}
#label15_TBinkop{left:600px;bottom:20px;}

/* GRIDVIEWS: */

/* Gridview_gridArtister: */

/* Templatefield-Kommentar (även för Gridview_grdResult): */
.cmt_align
{
    padding-left:30px;
}

/* Headerstyle-formatering: */
.header
{
    text-align:left;
}

.pos_gridartist
{
    padding-left:20px;
}

/* Gridview_grdResult: */

.pos_grdresult
{
    padding-left:20px;
}

```

```

/* grd_Artistdata: */

/* headerformatering: */

.pos_gridartistdata
{
    padding-left:20px;
}

.pos_kortbio
{
    padding-left:40px;
}

/*Gemensamt för alla tre gridviews: */
/* Item-formatering: */

.grid_style
{
    padding-left:20px;
}

/* Kortbiopanel och Kommentarpanel: */

.txtKortbio
{
    width:99%;
    height:300px;
    resize: none;
}

.lblArtistNo
{
    color:#006400;
    position:absolute;
    font-weight:bold;
    font-size:1.5em;

    bottom:248px;
    left:195px;
}

```



```

.txtArtistNo
{
  position:absolute;
  bottom:225px;
  left:193px;
}

.dynbtnKortbio
{
  position:absolute;
  bottom:220px;
  left:50px;
}

.dynbtnFetchKbio
{
  position:absolute;
  bottom:220px;
  left:90px;
}

.dynbtnRensaSkarm
{
  position:absolute;
  bottom:220px;
  left:130px;
}

/* CHARTS: */

/* 3D-Piechart till infosida: */

.ch3DPie
{
  position: relative;
  bottom:280px;
  left:270px;
}

.lbl3DPie
{
  position:relative;
  bottom:275px;
  right:60px;
  font-size:1em;
  font-family: Times New Roman, Georgia, Garamond, serif;
  font-weight:bold;
  color:#fff;
  background-color:#990033;
}

```

CSS

Fil som anpassar grafiska detaljer samt layout med mera i enlighet med den webbläsare som aktuell css-fil är anpassad efter. **Den här är för Internet Explorer.**

```
/*LAYERS*/

#container{
width: 712px;
height:640px;
position:absolute;
left:50%;
margin-left:-356px;
/*border: 5px solid #000;*/
}

#header
{
    height:40px;
    width:711px;
    background: transparent url(../bakgrundsbilder/header3.gif) no-repeat
center bottom;
    color:#fff;
    font-size:12px;
}

#content
{
    background:#D8D8D8;
    float:left;
    color:#000;
    font-size:12px;
    width:710px;
    height:421px;
}

#navigation
{
    background: transparent url(../bakgrundsbilder/nav_bakgrund3.gif) no-
repeat center bottom;
    color:#fff;
    font-size:12px;
    height:170px;
    width:712px;
}

/*DATUMVISNING*/
#datumruta
{
    position:absolute;
    right:50px;
    top:10px;
}
```

```

    font-weight:bold;
    color:#990033;
    background: #D8D8D8 no-repeat center bottom;
    width:121px;
    height:12px;
    border: 4px solid #fff;
    font-size:1.1em;
    font-family: Times New Roman, Georgia, Garamond, serif;
    padding-left:2px;
    padding-bottom:3px;
}

/*DISPLAY*/

.ruta{
    position:relative;
    width:662px;
    height:370px;
    border: 5px solid #000;
    background:#D8D8D8;
    float:left;
    margin-right:0px;
    margin-left:10px;
    margin-right:10px;
    margin-bottom:10px;
    margin-top:10px;
    padding:10px;
    overflow:auto;
    font-size:1em;
    font-family: Times New Roman, Georgia, Garamond, serif;

    /*Cool scroll-list: */
    scrollbar-arrow-color:#fff;
    scrollbar-base-color:#006400;
    scrollbar-track-color:#990033;
}

/*TEXTFORMATERING:*/

/*KORTBIO:*/

#visakortbiodata
{
    margin-right:240pt;
    color:#000;
}
/*#bredd{margin-right:280pt;}*/

/*KOMMENTAR:*/
#visakommentar

```

```

{
    margin-right:240pt;
    color:#000;
}

/*KORTBIOPANEL_INPUT:*/
#kortbiopanel_input
{
    margin-right:240pt;
    color:#000;
}

/*KOMMENTARPANEL_INPUT:*/
#kommentarpanel_input
{
    margin-right:240pt;
    color:#000;
}


/*DYNAMISKA BIOGRAFI BILDER*/

.dyn_pic
{
    border: 8px solid #990033;
    /*border: 8px solid black;*/
    display: block;
}


/*HÅLLER NAVRIGERINGSOBJEKTEN (KONTROLLYTAN) PÅ PLATS*/

.move{margin-left:-15.8%;}


/*LABELS-KONTROLLYTA-RAD 1*/

.labell1{color:#fff;bottom:152px;position:absolute;}

#labell1_Artist {left:150px;}

#label2_Album {left:280px;}

#label3_Format {left:410px;}

#label4_Press {left:540px;}

#label5_Kommentar {left:675px;}

```

```

#label6_ar {left:610px;}

/*TEXTUTOR-KONTROLLYTA-RAD 1*/

.text1{position:absolute;bottom:130px;}

#text3_Artist{left:150px;}
#text4_Album{left:280px;}
#text5_Format{left:410px;}
#text6_Press{left:540px;}
#text7_Kommentar{left:675px;}
#text8_ar{left:610px;}

/*LABELS-KONTROLLYTA-RAD 2*/

.label2{color:#fff;bottom:105px;position:absolute;}

#label7_VPfr {left:150px;}
#label8_VPtill {left:215px;}
#label9_Ursland {left:285px;}
#label10_Ursstad {left:352px;}
#label11_Musiktyp{left:430px;}
#label12_Kortbio{left:525px;}
#label13_Inm_dat{left:603px;}
#label14_Kop_grad{left:680px;}
#label15_Kop_kat{left:720px;}
#label16_Ca_pris{left:775px;}

/*TEXTUTOR-KONTROLLYTA-RAD 2*/

.text2{position:absolute;bottom:85px;}

```

```

#text9_VPfr{left:150px;}

#text10_VPtill{left:215px;}

#text11_Ursland{left:285px;}

#text12_Ursstad{left:352px;}

#text13_Musiktyp{left:430px;}

#text14_Kortbio{left:525px;}

#text15_Inm_dat{left:603px;}

#text16_Kop_grad{left:680px;}

#text17_Kop_kat{left:720px;}

#text18_Ca_pris{left:775px;}


/*KNAPPPAR-KONTROLLYTA*/

.input
{
    font-style: normal;
    font-weight:bold;
    color: #006400;
    font-size:0.8em;
    font-family: Times New Roman, Georgia, Garamond, serif;
    height:29px;
    background: transparent url(../knappar/knapp_nav.gif) no-repeat center
bottom;
    border:0;
    width:80px;
    position:absolute;
}

input.input:hover{color:#990033;}


/*DYNAMISK GEN KOMMENTAR-KNAPP*/

.cmt{
    background: transparent url(../knappar/tryck_knapp.gif) no-repeat
center bottom;
    border:0;
    width:70px;
    height:33px;
    color:#fff;
    font-weight:bold;
    font-size:6pt;
    font-size:0.8em;
    font-family: Times New Roman, Georgia, Garamond, serif;
}

```

```

input.cmt:hover{color:#fff; background: transparent
url(..knappar/tryck_knapp_hover.gif) no-repeat center bottom;}

/*DYNAMISK GEN ARTIST-KNAPP*/

.arti
{
    background: transparent url(..knappar/knapp_artist3.gif) no-repeat
center bottom;
    border:0;
    width:130px;
    height:20px;
    color:#990033;
    font-size:0.8em;
    font-family: Times New Roman, Georgia, Garamond, serif;
}

input.arti:hover{color:#006400; background: transparent
url(..knappar/knapp_artist3_hover.gif) no-repeat center bottom;}

/*RAD 1*/

.rad1{bottom:52px;}

#button_visa_databas{left:150px;}

#button_lagg_till{left:230px;}

#button_ta_bort{left:310px;}

#button_uppdatera{left:390px;}

#button_sok{left:470px;}

/*RAD 2*/

.rad2{bottom:27px;}

#btn21{left:150px;}

#btn22{left:230px;}

#btn23{left:310px;}

#btn24{left:390px;}

#btn25{left:470px;}

```

```

/*GENERELL FORMATERING*/

body
{
    background-color:#863E58;
    background-image:url(../bakgrundsbilder/bg_toning2.gif);
    background-repeat: repeat-x;

    /* Antal varianter på text-layout: */
    /*font-family: verdana, arial, sans-serif;*/
    /*font-family: Times New Roman, verdana, sans-serif;*/
    /*font-family: Times New Roman, Georgia, Garamond, serif;*/
    font-family:Garamond, Times New Roman, Georgia, serif;

}

a:link {color:#000; text-decoration: underline; }
a:active {color:#000; text-decoration: underline; }
a:visited {color:#000; text-decoration: underline; }
a:hover {color:#990033; text-decoration: underline; }

H1 {font-size:16pt;color:#fff;background-color:#990033;}
H2 {font-size:13pt; color:#000;}
H3 {font-size:10pt; color:#006400;}


/* VALKNAPPAR (Radio-buttons): */

.optSkivtabell
{
    position:absolute;
    bottom:60px;
    left:460px;
}

.optArtisttabell
{
    position:absolute;
    bottom:40px;
    left:460px;
}

.optInkopstabell
{
    position:absolute;
    bottom:20px;
    left:460px;
}

/* LABELS TILL VALKNAPPAR (Radio-buttons): */

```



```

.label13{color:#fff;position:absolute;font-weight:bold;font-size:1em;}

#label13_TBskivor{left:600px;bottom:60px;}

#label14_TBartist{left:600px;bottom:40px;}

#label15_TBinkop{left:600px;bottom:20px;}


/* GRIDVIEWS: */


/* Gridview_gridArtister: */


/* Templatefield-Kommentar (även för Gridview_grdResult): */

.cmt_align
{
    padding-left:30px;
}

/* Headerstyle-formatering: */

.header
{
    text-align:left;
}

.pos_gridartist
{
    padding-left:20px;
}


/* Gridview_grdResult: */

.pos_grdresult
{
    padding-left:20px;
}

```

```

/* grd_Artistdata: */

/* headerformatering: */

.pos_gridartistdata
{
    padding-left:20px;
}

.pos_kortbio
{
    padding-left:40px;
}


/*Gemensamt för alla tre gridviews: */

/* Item-formatering: */

.grid_style
{
    padding-left:20px;
}


/* Kortbiopanel och Kommentarpanel: */

.txtKortbio
{
    width:99%;
    height:300px;
    resize: none;
    overflow:auto;
}

.lblArtistNo
{
    color:#006400;
    position:absolute;
    font-weight:bold;
    font-size:1.5em;

    bottom:53px;
}

```

```

    left:195px;
}

.txtArtistNo
{
    position:absolute;
    bottom:30px;
    left:193px;
}

.dynbtnKortbio
{
    position:absolute;
    bottom:25px;
    left:50px;
}

.dynbtnFetchKbio
{
    position:absolute;
    bottom:25px;
    left:90px;
}

.dynbtnRensaSkarm
{
    position:absolute;
    bottom:25px;
    left:130px;
}


/* CHARTS: */

/* 3D-Piechart till infosida: */

.ch3DPie
{
    position: relative;
    bottom:280px;
    left:270px;
}

.lbl3DPie
{
    position:relative;
    bottom:275px;
    right:60px;
    font-size:1em;
    font-family: Times New Roman, Georgia, Garamond, serif;
    font-weight:bold;
    color:#fff;
    background-color:#990033;
}

```

CSS

Fil som anpassar grafiska detaljer samt layout med mera i enlighet med den webbläsare som aktuell css-fil är anpassad efter. **Den här är för Mozilla Firefox.**

```
/*LAYERS*/

#container
{
    width:712px;
    height:640px;
    position:absolute;
    left:50%;
    margin-left:-356px;
}

#header
{
    height:40px;
    width:711px;
    background: transparent url(../bakgrundsbilder/header3.gif) no-
repeat center bottom;
    float:left;
    color:#fff;
    font-size:12px;
}

#content
{
    background:#D8D8D8;
    float:left;
    color:#000;
    font-size:12px;
    width:711px;
    height:421px;
}

#navigation
{
    background: transparent url(../bakgrundsbilder/nav_bakgrund3.gif)
no-repeat center bottom;
    color:#fff;
    font-size:12px;
    height:170px;
    width:712px;
    float:left;
}

/*DATUMVISNING*/
#datumruta
{
    position:absolute;
    right:50px;
    top:10px;
    font-weight:bold;
}
```

```

        color:#990033;
        background: #D8D8D8 no-repeat center bottom;
        width:121px;
        height:12px;
        border: 4px solid #fff;
        font-size:1.1em;
        font-family: Times New Roman, Georgia, Garamond, serif;
        padding-left:2px;
        padding-bottom:3px;
    }

/*DISPLAY*/

.ruta{
    width:662px;
    height:370px;
    border: 5px solid #000;
    background:#D8D8D8;
    float:left;
    margin-right:0px;
    margin-left:10px;
    margin-right:10px;
    margin-bottom:10px;
    margin-top:10px;
    padding:10px;
    overflow:auto;
    font-size:1em;
    font-family: Times New Roman, Georgia, Garamond, serif;
}

/*TEXTFORMATERING:*/

/*KORTBIO:*/

#visakortbiodata
{
    margin-right:240pt;
    color:#000;
}
/*#bredd{margin-right:280pt;}*/

/*KOMMENTAR:*/
#visakommentar
{
    margin-right:240pt;
    color:#000;
}

/*KORTBIOPANEL_INPUT:*/
#kortbiopanel_input
{

```

```

        margin-right:240pt;
        color:#000;
    }

/*KOMMENTARPANEL_INPUT:*/
#kommentarpanel_input
{
    margin-right:240pt;
    color:#000;
}


/*DYNAMISKA BIOGRAFI BILDER*/

.dyn_pic
{
    border: 8px solid #990033;
    display: block;
}


/*HÅLLER NAVERIGERINGSOBJEKTEN (KONTROLLYTAN) PÅ PLATS*/

.move{margin-left:-15.8%;}


/*LABELS-KONTROLLYTA-RAD 1*/

.label1{color:#fff;bottom:152px;position:absolute;}

#label1_Artist {left:150px;}
#label2_Album {left:280px;}
#label3_Format {left:410px;}
#label4_Press {left:540px;}
#label5_Kommentar {left:675px;}
#label6_ar {left:610px;}


/*TEXTRUTOR-KONTROLLYTA-RAD 1*/

```

```

.text1{position:absolute;bottom:130px;}

#text3_Artist{left:150px;}
#text4_Album{left:280px;}
#text5_Format{left:410px;}
#text6_Press{left:540px;}
#text7_Kommentar{left:675px;}
#text8_ar{left:610px;}

/*LABELS-KONTROLLYTA-RAD 2*/

.label2{color:#fff;bottom:105px;position:absolute;}

#label7_VPfr {left:150px;}
#label8_VPtill {left:215px;}
#label9_Ursland {left:285px;}
#label10_Ursstad {left:352px;}
#label11_Musiktyp{left:422px;}
#label12_Kortbio{left:525px;}
#label13_Inm_dat{left:595px;}
#label14_Kop_grad{left:668px;}
#label15_Kop_kat{left:710px;}
#label16_Ca_pris{left:770px;}

/*TEXTRUTOR-KONTROLLYTA-RAD 2*/

.text2{position:absolute;bottom:85px;}

#text9_VPfr{left:150px;}
#text10_VPtill{left:215px;}
#text11_Ursland{left:285px;}
#text12_Ursstad{left:352px;}

```

```

#text13_Musiktyp{left:422px;}

#text14_Kortbio{left:525px;}

#text15_Inm_dat{left:595px;}

#text16_Kop_grad{left:668px;}

#text17_Kop_kat{left:710px;}

#text18_Ca_pris{left:770px;}


/*KNAPPPAR-KONTROLLYTA*/

.input
{
    font-style: normal;
    font-weight:bold;
    color: #006400;
    font-size:0.8em;
    font-family: Times New Roman, Georgia, Garamond, serif;
    height:29px;
    background: transparent url(../knappar/knapp_nav.gif) no-repeat center
bottom;
    border:0;
    width:80px;
    position:absolute;
}

input.input:hover{color:#990033;}


/*DYNAMISK GEN KOMMENTAR-KNAPP*/

.cmt
{
    background: transparent url(../knappar/tryck_knapp.gif) no-repeat
center bottom;
    border:0;
    width:70px;
    height:33px;
    color:#fff;
    font-weight:bold;
    font-size:0.8em;
    font-family: Times New Roman, Georgia, Garamond, serif;
}

input.cmt:hover{color:#fff; background: transparent
url(../knappar/tryck_knapp_hover.gif) no-repeat center bottom;}


/*DYNAMISK GEN ARTIST-KNAPP*/

.arti

```



```

{
    background: transparent url(../knappar/knapp_artist3.gif) no-repeat
center bottom;
    border:0;
    width:130px;
    height:20px;
    color:#990033;
    font-size:0.8em;
    font-family: Times New Roman, Georgia, Garamond, serif;
}

input.arti:hover{color:#006400; background: transparent
url(../knappar/knapp_artist3_hover.gif) no-repeat center bottom;}

```

```

/*RAD 1*/

.rad1{bottom:52px;}

#button_visa_databas{left:150px;}

#button_lagg_till{left:230px;}

#button_ta_bort{left:310px;}

#button_uppdatera{left:390px;}

#button_sok{left:470px;}

```

```

/*RAD 2*/

.rad2{bottom:27px;}

#btn21{left:150px;}

#btn22{left:230px;}

#btn23{left:310px;}

#btn24{left:390px;}

#btn25{left:470px;}

```

```

/*GENERELL FORMATERING*/

body
{
    background-color:#863E58;

```

```

background-image:url(../bakgrundsbilder/bg_toning2.gif);
background-repeat: repeat-x;

/* Antal varianter på text-layout: */
/*font-family: verdana, arial, sans-serif;*/
/*font-family: Times New Roman, verdana, sans-serif;*/
/*font-family: Times New Roman, Georgia, Garamond, serif;*/
font-family:Garamond, Times New Roman, Georgia, serif;

}

a:link {color:#000; text-decoration: underline; }
a:active {color:#000; text-decoration: underline; }
a:visited {color:#000; text-decoration: underline; }
a:hover {color:#990033; text-decoration: underline; }

H1 {font-size:16pt;color:#fff;background-color:#990033;}
H2 {font-size:13pt; color:#000;}
H3 {font-size:10pt; color:#006400;}


/* VALKNAPPAR (Radio-buttons): */

.optSkivtabell
{
    position:absolute;
    bottom:60px;
    left:460px;
}

.optArtisttabell
{
    position:absolute;
    bottom:40px;
    left:460px;
}

.optInkopstabell
{
    position:absolute;
    bottom:20px;
    left:460px;
}

/* LABELS TILL VALKNAPPAR (Radio-buttons): */

.label13{color:#fff;position:absolute;font-weight:bold;font-size:1em;}

#label13_TBskivor{left:600px;bottom:60px;}

#label14_TBartist{left:600px;bottom:40px;}

#label15_TBinkop{left:600px;bottom:20px;}

```

```

/* GRIDVIEWS: */

/* Gridview_gridArtist: */

/* Templatefield-Kommentar (även för Gridview_grdResult): */
.cmt_align
{
    padding-left:30px;
}

/* Headerstyle-formatering: */
.header
{
    text-align:left;
}

.pos_gridartist
{
    padding-left:20px;
}


/* Gridview_grdResult: */

.pos_grdresult
{
    padding-left:20px;
}


/* grd_Artistdata: */

```

```

/* headerformatering: */

.pos_gridartistdata
{
    padding-left:20px;
}

.pos_kortbio
{
    padding-left:40px;
}

/*Gemensamt för alla tre gridviews: */

/* Item-formatering: */

.grid_style
{
    padding-left:20px;
}

/* Kortbiopanel och Kommentarpanel: */

.txtKortbio
{
    width:99%;
    height:300px;
    resize: none;
}

.lblArtistNo
{
    color:#006400;
    position:absolute;
    font-weight:bold;
    font-size:1.5em;

    bottom:248px;
    left:195px;
}

.txtArtistNo
{
    position:absolute;
    bottom:225px;
    left:193px;
}

```

```

}

.dynbtnKortbio
{
    position:absolute;
    bottom:220px;
    left:50px;
}

.dynbtnFetchKbio
{
    position:absolute;
    bottom:220px;
    left:90px;
}

.dynbtnRensaSkarm
{
    position:absolute;
    bottom:220px;
    left:130px;
}


/* CHARTS: */

/* 3D-Piechart till infosida: */

.ch3DPie
{
    position: relative;
    bottom:280px;
    left:270px;
}

.lbl3DPie
{
    position:relative;
    bottom:275px;
    right:60px;
    font-size:1em;
    font-family: Times New Roman, Georgia, Garamond, serif;
    font-weight:bold;
    color:#fff;
    background-color:#990033;
}

```

MELLANSKIKTET

Mellanskiktet – Komponent: DB_Callcode

Olika klasser med funktionaliteter för Skivtabellen

Klass som raderar innehållet i alla textrutor och dropdownmenyer

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class Borja_om
    {
        Page sida = new Page();

        //Raderar innehållet i alla textrutor och dropdownmenyer:
        public void button_Borja_om()
        {
            Control myForm = sida.FindControl("Form1");
            foreach (Control ctl in myForm.Controls)
            {
                if
                (ctl.GetType().ToString().Equals("System.Web.UI.WebControls.TextBox"))
                ((TextBox)ctl).Text = "";
            }
        }
    }
}
```


Klass som hanterar kryssrutor för gridview gridArtister och grdResult samt ser till att motsvarande poster i databasen är (av)markerade.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Drawing;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class chk_Gridviews
    {
        SetChks db = new SetChks();

        //Kryssruta för gridview gridArtister och grdResult:
        public void chkCDval_CheckedChanged(Gridview gridArtister,
        Gridview grdResult)
        {
            Gridview crossgrid = new Gridview();

            if (grdResult.Visible == true)
            {
                crossgrid = grdResult;
            }
            else if (gridArtister.Visible == true)
            {
                crossgrid = gridArtister;
            }

            foreach (GridviewRow gvRow in crossgrid.Rows)
            {
                DerivCheckBox chkssel =
                (DerivCheckBox)gvRow.FindControl("chkCDval");

                if (chkssel.Checked == true)
                {
                    Int32 val = 1;
                    Int32 row = Int32.Parse(chkssel.UnText);

                    //Sätter markeringsvärden för markerad rad:
                }
            }
        }
    }
}
```

```

        gvRow.BackColor =
System.Drawing.ColorTranslator.FromHtml("#738A9C");
        gvRow.ForeColor =
System.Drawing.ColorTranslator.FromHtml("#F7F7F7");
        gvRow.Font.Bold = true;

        try
        {
            db.UpdateRecord_ChkBox(row, val);
        }
        finally
        { }

    }
    else if (chkssel.Checked == false)
    {

        Int32 val = 0;
        Int32 row = Int32.Parse(chkssel.UnText);

        //Sätter tillbaka markeringsvärden för avmarkerad rad:
        gvRow.BackColor =
System.Drawing.ColorTranslator.FromHtml("#F7F7F7");
        gvRow.ForeColor =
System.Drawing.ColorTranslator.FromHtml("#4A3C8C");
        gvRow.Font.Bold = false;

        try
        {
            db.UpdateRecord_ChkBox(row, val);
        }
        finally
        { }

    }

}

}

}

```

Klass som hanterar de dynamiska knapparna för visning av kommentarer respektive artistdata.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class dbtn_KomArt
    {
        Clear clr = new Clear();
        Proc_act db = new Proc_act();

        //Kommentarknapp och Artistknapp:
        public void gridArtister_RowCommand(GridView grd_Artistdata,
            HtmlGenericControl display, HtmlGenericControl display2, Page sida,
            GridViewCommandEventArgs e)
        {
            if (e.CommandName == "Kommentar")
            {
                int row = Convert.ToInt32(e.CommandArgument);
                string kom = "";
                int rowcount = 0;

                try
                {
                    DataSet ds = new DataSet();
                    ds = db.CD_Kommentarer(row);
                    kom = ds.Tables[0].Rows[0]["Kommentar"].ToString();
                    rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());
                }
                finally
                { }

                if (rowcount > 0)
                {
                    //Rensar display från text och gridviews
                    clr.Clean_surfaces(sida);

                    DateTime Now = DateTime.Now;

                    string add = "<h1>Kommentar No: " + row + "</h1>";
                    add += "<div id=visakommentar>" + kom + "</div>";
                    add += "<h3>" + "Sidan skapades: " + Now + "</h3>";
                    display.InnerHtml = add;
                }
            }
        }
    }
}
```

```

    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }

}
else if (e.CommandName == "Artist")
{

    int row = Convert.ToInt32(e.CommandArgument);
    int rowcount = 0;

    try
    {
        DataSet ds = new DataSet();
        ds = db.CD_VisaArtistData(row);

        grd_Artistdata.DataSource = ds;
        grd_Artistdata.DataBind();

        rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());

    }
    finally
    { }

    if (rowcount > 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        grd_Artistdata.Visible = true;

        DateTime Now = DateTime.Now;

        string add1 = "<h1>Artistdata No: " + row + "</h1>";
        string add2 = "<h3>" + "Sidan skapades: " + Now +
"</h3>";

        display.InnerHtml=add1;
        display2.InnerHtml = add2;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}

```

}
}
}

Klass för inmatning av längre kommentarer till Skivtabell via kommentarpanel

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class dbtn_Kommentar_Input
    {
        Clear clr = new Clear();
        Proc_act db = new Proc_act();

        //Matar in längre kommentar till skivtabell:
        public void dynbtnKommentar(HtmlGenericControl display,
            TextBox txtKommentar, TextBox txtKommentarNo, Page sida)
        {
            int noaddkommentar = 0;

            try
            {
                noaddkommentar = db.AddKommentar(txtKommentar.Text,
Int32.Parse(txtKommentarNo.Text));
            }
            finally
            { }

            if (noaddkommentar > 0)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(sida);

                DateTime Now = DateTime.Now;

                string add = "<h1>Ny kommentar har lagts till
databasen</h1>";
                add += "<h2>Kommentar för artistno: " + txtKommentarNo.Text
+ "</h2>";
                add += "<h2>Inmatat till DB:</h2> ";
                add += "<div id=kommentarpanel_input>" + txtKommentar.Text
+ "</div>";

                add += "<h3>" + "Sidan skapades: " + Now + "</h3>";
            }
        }
    }
}
```

```

        display.InnerHtml = add;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}
}
}

```

Klass för hämtning av längre kommentar från skivtabell (via kommentarpanel)

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class dbtn_Kommentar_Output
    {
        Clear clr = new Clear();
        Proc_act db = new Proc_act();

        //Hämtar längre kommentar från skivtabell (via kommentarpanel)
        public void dynbtnKommentarFetch(HtmlGenericControl display,
            Panel pnlKommentar, TextBox txtKommentar, TextBox txtKommentarNo,
            Page sida)
        {
            int rowcount = 0;

            try
            {
                DataSet ds = new DataSet();

                ds = db.FetchKommentar(Int32.Parse(txtKommentarNo.Text));
                txtKommentar.Text =
ds.Tables[0].Rows[0]["Kommentar"].ToString();
                rowcount = int.Parse(ds.Tables[0].Rows.Count.ToString());
            }
            finally
            { }

            if (rowcount == 1)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(sida);

                pnlKommentar.Visible = true;
            }
            else
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(sida);

                display.InnerHtml = "<h3>Inga erhållna värden från
databas.</h3>";
            }
        }
    }
}
```


}
}
}

Klass för att visa en artist kortbiografi.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class dbtn_Visa_Kortbio
    {
        Clear clr = new Clear();
        Proc_act db = new Proc_act();

        //Visar en artists kortbiografi:
        public void grd_Artistdata_RowCommand(HtmlGenericControl display,
        GridViewCommandEventArgs e, Page sida)
        {
            if (e.CommandName == "Artist_kortbio")
            {
                string row = (string)e.CommandArgument;
                string kortbio = "";
                string artist = "";
                int rowcount = 0;

                try
                {
                    DataSet ds = new DataSet();
                    ds = db.CD_VisaKortBio(row);

                    kortbio = ds.Tables[0].Rows[0]["Kortbio"].ToString();
                    artist = ds.Tables[0].Rows[0]["Artist"].ToString();

                    rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());
                }
                finally
                { }

                if (rowcount > 0)
                {
```

```

//Rensar display från text och gridviews
clr.Clean_surfaces(sida);

DateTime Now = DateTime.Now;

string add = "<h1>Kort biografi om " + artist +
":</h1>";

add += "<div id=visakortbiodata>" + kortbio + "</div>";

display.InnerHtml = add;

}
else
{
//Rensar display från text och gridviews
clr.Clean_surfaces(sida);

display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
}
}
}
}
}
}
}

```

Klass för hämtning av en rad från Skivtabellen som skall uppdateras

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class Fetch
    {
        Clear clr = new Clear();
        Proc_act db = new Proc_act();

        //Hämtar en rad från aktuell tabell som skall uppdateras:
        public void button_fetch(Gridview gridArtister, Gridview grdResult,
            HtmlGenericControl display, HtmlGenericControl display2, Page sida,
            TextBox text3_Artist, TextBox text4_Album, DropDownList
text5_Format,
            DropDownList text6_Press, DropDownList text8_ar,
            TextBox text7_Kommentar)
        {
            Proc_act db = new Proc_act();

            Gridview fetchgrid = new Gridview();

            int a = 0;
            int rowcount = 0;

            if (grdResult.Visible == true)
            {
                fetchgrid = grdResult;
            }
            else if (gridArtister.Visible == true)
            {
                fetchgrid = gridArtister;
            }

            foreach (GridViewRow gvRow in fetchgrid.Rows)
            {
                DerivCheckBox chkse1 =
                (DerivCheckBox)gvRow.FindControl("chkCDval");

                if (chkse1.Checked == true)
```

```

        {
            a++;
        }
    }

    if (a == 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        display.InnerHtml = "<h2>Ingen kryssruta ifyllld.</h2>";
    }
    else if (a > 1)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        display.InnerHtml = "<h2>Endast EN kryssruta får vara
ifylld.</h2>";
    }
    else
    {
        try
        {
            DataSet ds = new DataSet();
            ds = db.FetchRecord();

            grdResult.DataSource = ds;
            grdResult.DataBind();

            text3_Artist.Text =
ds.Tables[0].Rows[0]["Artist_no"].ToString();
            text4_Album.Text =
ds.Tables[0].Rows[0]["Album"].ToString();
            text5_Format.Text =
ds.Tables[0].Rows[0]["Format"].ToString();
            text6_Press.Text =
ds.Tables[0].Rows[0]["Press"].ToString();
            text8_ar.Text = ds.Tables[0].Rows[0]["Ar"].ToString();
            text7_Kommentar.Text =
ds.Tables[0].Rows[0]["Kommentar"].ToString();

            rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());
        }
        finally
        { }

        if (rowcount == 1)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(sida);

            grdResult.HeaderRow.FindControl("chkCDval_ALL").Visible
= false;

            grdResult.Visible = true;

```

```

        DateTime Now = DateTime.Now;

        string add1 = "<h1>Post har hämtats till databas för
uppdatering</h1>";
        string add2 = "<h3>" + "Sidan skapades: " + Now +
"</h3>";

        display.InnerHtml = add1;
        display2.InnerHtml = add2;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}
}
}
}
}

```

Klass för hantering av headerkryssruta för gridview grdResult.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class hchk_grdResult
    {
        SetChks db = new SetChks();

        //Headerkryssruta för gridview grdResult:
        public void chkCDvalALLGrdRes_CheckedChanged(GridView grdResult,
            object sender)
        {
            foreach (GridViewRow gvRow in grdResult.Rows)
            {
                DerivCheckBox chkssel =
                (DerivCheckBox) (gvRow.Cells[0].FindControl("chkCDval"));
                chkssel.Checked = ((DerivCheckBox) sender).Checked;

                if (chkssel.Checked == true)
                {
                    Int32 val = 1;
                    Int32 row = Int32.Parse(chkssel.UnText);

                    try
                    {
                        db.UpdateRecord_ChkBox(row, val);
                    }
                    finally
                    { }
                }
                else if (chkssel.Checked == false)
                {
                    Int32 val = 0;
                    Int32 row = Int32.Parse(chkssel.UnText);

                    try
                    {
                        db.UpdateRecord_ChkBox(row, val);
                    }
                }
            }
        }
    }
}
```

```
        }  
        finally  
        { }  
    }  
}  
  
}
```


Klass för hantering av headerkryssruta för gridview gridArtister.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class hchk_gridArtister
    {
        SetChks db = new SetChks();

        //Headerkryssruta för gridview gridArtister:
        public void chkCDvalALLMain_CheckedChanged(GridView
gridArtister, object sender)
        {
            foreach (GridViewRow gvRow in gridArtister.Rows)
            {
                DerivCheckBox chkssel =
(DerivCheckBox) (gvRow.Cells[0].FindControl("chkCDval"));
                chkssel.Checked = ((DerivCheckBox) sender).Checked;

                if (chkssel.Checked == true)
                {
                    Int32 val = 1;

                    try
                    {
                        db.UpdateRecord_ChkBox(val);
                    }
                    finally
                    { }
                }
                else if (chkssel.Checked == false)
                {
                    Int32 val = 0;

                    try
                    {
                        db.UpdateRecord_ChkBox(val);
                    }
                    finally

```

```

    {
      }
    }
  }
}
{ }
```

Klass för hantering av information om aktuell data i databas

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Collections.Specialized;
using System.Collections;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class Info
    {
        Clear clr = new Clear();
        Proc_act db = new Proc_act();

        //Ger information om aktuell data i databas:
        public void button_info(HtmlGenericControl display,
            HtmlGenericControl display2,
            Page sida)
        {
            OrderedDictionary dictRec = new OrderedDictionary();
            int rowcount = 0;

            try
            {
                dictRec = db.GetInfo();
                rowcount = dictRec.Count;
            }
            finally
            { }

            //Value- och key-kollektioner initieras mha Interface:
            ICollection keyKollektion = dictRec.Keys;
            ICollection valueKollektion = dictRec.Values;

            //Skapar arrays och kopierar kollektionerna till desamma:
            String[] keys = new String[rowcount];
            int[] values = new int[rowcount];
            keyKollektion.CopyTo(keys, 0);
            valueKollektion.CopyTo(values, 0);

            if (rowcount > 0)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(sida);
            }
        }
    }
}
```

```

DateTime Now = DateTime.Now;

//Skapar de olika tabellerna:
Table tblCDDVD = new Table();
Table tblVinyl = new Table();
Table tblDataMeida = new Table();
Table tblKluster = new Table();

int sum_cddvd = 0;
int sum_vinyl = 0;
int sum_datamedia = 0;

//Rubrik för infosidan:
string add = "<h1>Följande info för tabeller finns: </h1>";
display.InnerHtml = add;

//Skriver ut CD/DVD-info om skivtabell:
//Rubrikrad:
TableRow rubrikrad_cddvd = new TableRow();
tblCDDVD.Controls.Add(rubrikrad_cddvd);

TableHeaderCell hcell_cddvd = new TableHeaderCell();
//hcell_cddvd.BorderWidth = 1;
hcell_cddvd.HorizontalAlign = HorizontalAlign.Left;
hcell_cddvd.Text = "CD/DVD:";
rubrikrad_cddvd.Controls.Add(hcell_cddvd);

//Tabellinnehåll:
for (int i = 0; i < 11; i++)
{
    TableRow rad = new TableRow();
    tblCDDVD.Controls.Add(rad);

    TableCell cell = new TableCell();
    //cell.BorderWidth = 1;
    cell.Text = "Antal " + keys[i] + ": " + "<b>" +
values[i] + "</b>";
    rad.Controls.Add(cell);
    sum_cddvd+=values[i];
}

//Summeringsrad:
TableRow sumrad_cddvd = new TableRow();

TableHeaderCell hsumma_cddvd = new TableHeaderCell();
hsumma_cddvd.HorizontalAlign = HorizontalAlign.Left;
hsumma_cddvd.Text = "Summa: " + sum_cddvd.ToString() + "
cd/dvd(s) ";

sumrad_cddvd.Controls.Add(hsumma_cddvd);
tblCDDVD.Controls.Add(sumrad_cddvd);

//Skriver ut vinyl-info om skivtabell:
//Rubrikrad:
TableRow rubrikrad_vinyl = new TableRow();
tblVinyl.Controls.Add(rubrikrad_vinyl);

TableHeaderCell hcell_vinyl = new TableHeaderCell();

```

```

//hcell_vinyl.BorderWidth = 1;
hcell_vinyl.HorizontalAlign = HorizontalAlign.Left;
hcell_vinyl.Text = "Vinyl:";
rubrikrad_vinyl.Controls.Add(hcell_vinyl);

//Tabellinnehåll:
for (int i = 11; i < 23; i++)
{
    TableRow rad = new TableRow();
    tblVinyl.Controls.Add(rad);

    TableCell cell = new TableCell();
    //cell.BorderWidth = 1;
    cell.Text = "Antal " + keys[i] + ": " + "<b>" +
values[i] + "</b>";
    rad.Controls.Add(cell);
    sum_vinyl += values[i];
}

//Summeringsrad:
TableRow sumrad_vinyl = new TableRow();

TableHeaderCell hsumma_vinyl = new TableHeaderCell();
hsumma_vinyl.HorizontalAlign = HorizontalAlign.Left;
hsumma_vinyl.Text = "Summa: " + sum_vinyl.ToString() + "
vinyl(er)";

sumrad_vinyl.Controls.Add(hsumma_vinyl);
tblVinyl.Controls.Add(sumrad_vinyl);

//Skriver ut datamedia-info om skivtabell:
//Rubrikrad:
TableRow rubrikrad_datam = new TableRow();
tblDataMeida.Controls.Add(rubrikrad_datam);

TableHeaderCell hcell_datam = new TableHeaderCell();
//hcell_datam.BorderWidth = 1;
hcell_datam.HorizontalAlign = HorizontalAlign.Left;
hcell_datam.Text = "Datamedia:";
rubrikrad_datam.Controls.Add(hcell_datam);

//Tabellinnehåll:
//for (int i = 11; i < 23; i++)
//{
TableRow rad_datam = new TableRow();
tblDataMeida.Controls.Add(rad_datam);

    TableCell cell_datam = new TableCell();
    //cell_datam.BorderWidth = 1;
    cell_datam.Text = "Antal " + keys[23] + ": " + "<b>" +
values[23] + "</b>";
    rad_datam.Controls.Add(cell_datam);
    sum_datamedia += values[23];
//}

//Summeringsrad:

```

```

        TableRow sumrad_datamedia = new TableRow();

        TableHeaderCell hsumma_datamedia = new TableHeaderCell();
        hsumma_datamedia.HorizontalAlign = HorizontalAlign.Left;
        hsumma_datamedia.Text = "Summa: " +
sum_datamedia.ToString() + " mp3(s)";
        sumrad_datamedia.Controls.Add(hsumma_datamedia);
        tblDataMeida.Controls.Add(sumrad_datamedia);


        //Sätter upp layouten för ovanstående tabeller:
        TableRow rad_Kluster = new TableRow();

        TableCell cell11_Kluster = new TableCell();
        TableCell cell12_Kluster = new TableCell();
        TableCell cell13_Kluster = new TableCell();

        cell11_Kluster.Controls.Add(tblCDDVD);
        cell12_Kluster.Controls.Add(tblVinyl);
        cell13_Kluster.Controls.Add(tblDataMeida);

        cell11_Kluster.VerticalAlign = VerticalAlign.Top;
        cell12_Kluster.VerticalAlign = VerticalAlign.Top;
        cell13_Kluster.VerticalAlign = VerticalAlign.Top;

        rad_Kluster.Controls.Add(cell11_Kluster);
        rad_Kluster.Controls.Add(cell12_Kluster);
        rad_Kluster.Controls.Add(cell13_Kluster);

        tblKluster.Controls.Add(rad_Kluster);


        //Visar alla tabeller inkl. layout på displayen:
        display.Controls.Add(tblKluster);

        //Totalsumma för antal skivor i skivtabell:
        int totalsumma = sum_cddvd + sum_vinyl + sum_datamedia;

        //Visar tid när sidan skapades och totalsumma för
skivdatabas:
        string add2 = "<h3 id=nogreen> Totalsumma för skivdatabas: "
+ totalsumma + " enheter </h3>";
        add2 += "<h3>" + "Sidan skapades: " + Now + "</h3>";
        display2.InnerHtml = add2;

        //Visar 3D-piechart + dess label:
        //ch3DPie.Visible = true;
        //lbl3DPie.Visible = true;

    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);
    }
}

```

```
        display.InnerHtml = "<h2>Inga erhållna värden från  
databas.</h2>";  
    }  
  
    }  
  
}
```

Klass för tillägg av post i Skivtabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Collections;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class Lagg_till
    {
        Clear clr = new Clear();
        Proc_act db = new Proc_act();

        //Lägger till data i databas
        public void button_lagg_till(HtmlGenericControl display,
            TextBox text3_Artist, TextBox text4_Album,
            DropDownList text5_Format, DropDownList text6_Press,
            DropDownList text8_ar, TextBox text7_Kommentar,
            BaseValidator vldArtistOnlyNum, BaseValidator vldArtistNotEmpty,
            BaseValidator vldArtistValidRange, Page sida)
        {
            Int32 val = 0;
            Int32 noaddrecs = 0;
            string errorsum = "";
            int rowcount = 0;

            string vldArtistValidRangeErrText="<h2>Artistnummer utanför
registrerade värden ";
            vldArtistValidRangeErrText+= "i databas. Kolla i artisttabellen
och försök igen.</h2>";

            if (!sida.IsValid)
            {
                Dictionary<BaseValidator, string> dict = new
Dictionary<BaseValidator, string>();
                dict.Add(vldArtistNotEmpty, "<h2>Ej matat in artistnummer.
Kolla i artisttabellen och försök igen.</h2>");
                dict.Add(vldArtistOnlyNum, "<h2>Endast siffor är tillåtna
som artistnummer.</h2>");
                dict.Add(vldArtistValidRange, vldArtistValidRangeErrText);

                rowcount = dict.Count;

                //Value- och key-kollektioner initieras mha Interface:
                ICollection keyKollektion = dict.Keys;
```



```

ICollection valueKollektion = dict.Values;

//Skapar arrays och kopierar kollektionerna till desamma:
BaseValidator[] keys = new BaseValidator[rowcount];
string[] values = new string[rowcount];
keyKollektion.CopyTo(keys, 0);
valueKollektion.CopyTo(values, 0);

//Utvärderar fält mha validationskontroller och skriver
//felmeddelande till gemensam sträng:
for (int i = 0; i < rowcount; i++)
{
    if (!keys[i].IsValid)
    {
        errorsum += values[i];
    }
}

//Rensar display från text och gridviews
clr.Clean_surfaces(sida);

//Skriver gemensam felmeddelande-sträng till display:
display.InnerHtml = errorsum;
}
else
{
    try
    {
        noaddrescs = db.AddRecord(val,
Int32.Parse(text3_Artist.Text),
        text4_Album.Text, text5_Format.Text, text6_Press.Text,
text8_ar.Text,
        text7_Kommentar.Text);
    }
    finally
    { }

    if (noaddrescs > 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        DateTime Now = DateTime.Now;

        string add = "<h1>Ny post har lagts till
databasen</h1>";

        add += "<h2>Data:</h2>";
        add += "<h2><div id=visakommentar>KryssruteVärde = " +
val + "<br />";

        add += "Artist_no = " + text3_Artist.Text + "<br />";
        add += "Album = " + text4_Album.Text + "<br />";
        add += "Format = " + text5_Format.Text + "<br />";
        add += "Press = " + text6_Press.Text + "<br />";
        add += "År = " + text8_ar.Text + "<br />";
        add += "Kommentar (se nedan): </div></h2>";
        add += "<div id=visakommentar>" + text7_Kommentar.Text
+ "</div>";

        add += "<h3>" + "Sidan skapades: " + Now + "</h3>";

```

```

        display.InnerHtml = add;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}
}
}

```

Klass som fungerar som en omstart och rensar alla inmatningsrutor odyl + display.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class Refresh_TB
    {
        //Fungerar som en omstart och rensar alla inmatningsrutor odyl +
display:
        public void button_Refresh_TB(HttpServerUtility server, string
retursida)
        {

            SetChks db = new SetChks();
            Int32 val = 0;

            try
            {
                db.UpdateRecord_ChkBox(val);
                server.Transfer(retursida);
            }
            finally
            { }

        }

    }
}
```

Klass som söker genom Skivtabellen enligt givna kriterier.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{

    public class Sok
    {
        Clear clr = new Clear();
        Proc_act db = new Proc_act();

        DataSet ds = new DataSet();

        //Tar emot angivelsen av hur lång tid cache:n skall verka
        private double delay;
        public double Delay
        { set { delay = value; }}

        //Söker i databas:
        public void button_sok(GridView grdResult, Page sida,
            HtmlGenericControl display, TextBox text3_Artist,
            TextBox text4_Album, DropDownList text5_Format,
            DropDownList text6_Press, DropDownList text8_ar,
            TextBox text7_Kommentar)
        {
            int rowcount = 0;

            string artist = text3_Artist.Text;
            string album = text4_Album.Text;
            string format = text5_Format.Text;
            string press = text6_Press.Text;
            string ar = text8_ar.Text;
            string kommentar = text7_Kommentar.Text;

            string res = "";
            string cnt = "";

            //Använder alla sökkriterier för att generera unika
            cachenycklar:
            ds = (DataSet)HttpRuntime.Cache["cache" + artist + album +
            format + press
                + ar + kommentar];

            if (ds == null)
```

```

    {
        try
        {
            ds = db.SearchRecords(artist, album, format, press, ar,
kommentar);

            //Cache för varje unik sökning + 20 sek timeout (värdet
av delay):
            HttpRuntime.Cache.Insert("cache" + artist + album +
format + press + ar + kommentar, ds,
            null, DateTime.Now.AddSeconds(delay), TimeSpan.Zero);

            cnt = HttpRuntime.Cache.Count.ToString();
            res = "Skapad och adderad till cache.";
        }
        finally
        { }
    }
    else
    {
        cnt = HttpRuntime.Cache.Count.ToString();
        res = "Hämtad från cache.";
    }

    grdResult.DataSource = ds;
    grdResult.DataBind();

    //Kollar så dataset har innehåll + exekvering är ok:
    rowcount = ds.Tables[0].Rows.Count;

    if (rowcount > 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        grdResult.Visible = true;

        DateTime Now = DateTime.Now;

        string add = "<h1>" + rowcount + " Post(er)
hittades:</h1>";

        add += "<h2>" + "Cache status: " + res + "<br />";
        add += "Antal objekt i cache: " + cnt + "<br />";
        add += "Delay status: " + delay + "</h2>";

        add += "<h3>" + "Sidan skapades: " + Now + "</h3>";

        display.InnerHtml = add;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}

```

}

}
}

Klass för att ta bort post(er) i Skivtabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class Ta_bort
    {
        Clear clr = new Clear();

        //Tar bort data från databas
        public void button_ta_bort(GridView gridArtister, GridView
grdResult,
        HtmlGenericControl display, Page sida)
        {
            Proc_act db = new Proc_act();

            Int32 val = 1;
            Int32 nodelrecs = 0;
            Int32 a = 0;

            GridView delgrid = new GridView();

            if (grdResult.Visible == true)
            {
                delgrid = grdResult;
            }
            else if (gridArtister.Visible == true)
            {
                delgrid = gridArtister;
            }

            foreach (GridViewRow gvRow in delgrid.Rows)
            {
                DerivCheckBox chkssel =
(DerivCheckBox)gvRow.FindControl("chkCDval");
```

```

        if (chkssel.Checked == true)
        {
            a++;
        }
    }

    if (a == 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces(sida);

        string mess="<h2>Ej kryssat för någon ruta. ";
        mess += "Inget kan därmed tas bort. Kryssa för minst en
ruta</h2>";
        display.InnerHtml = mess;
    }
    else if (a >= 1)
    {
        try
        {
            nodelrecs = db.DelRecord(val);
        }
        finally
        { }

        if (nodelrecs > 0)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(sida);

            DateTime Now = DateTime.Now;

            string add = "<h1>" + nodelrecs + " Post(er) har tagits
bort från databasen</h1>";
            add += "<h3>" + "Sidan skapades: " + Now + "</h3>";
            display.InnerHtml = add;
        }
        else
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(sida);

            display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
        }
    }
}

}
}

```


Klass som uppdaterar Skivtabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class Uppdatera
    {
        Clear clr = new Clear();

        //Uppdaterar databas:
        public void button_oppdatera(Gridview gridArtister, Gridview
grdResult,
        HtmlGenericControl display, Page sida, TextBox text3_Artist,
        TextBox text4_Album, DropDownList text5_Format, DropDownList
text6_Press,
        DropDownList text8_ar, TextBox text7_Kommentar)
        {
            Int32 noupdrecs = 0;
            Gridview updategrid = new Gridview();

            if (grdResult.Visible == true)
            {
                updategrid = grdResult;
            }
            else if (gridArtister.Visible == true)
            {
                updategrid = gridArtister;
            }

            foreach (GridViewRow gvRow in updategrid.Rows)
            {
                DerivCheckBox chkse1 =
(DerivCheckBox) gvRow.FindControl("chkCDval");

                if (chkse1.Checked == true)
                {
                    Proc_act db = new Proc_act();
```

```

Int32 row = Int32.Parse(chksel.UnText);
Int32 artist_no = Int32.Parse(text3_Artist.Text);
string album = text4_Album.Text;
string format = text5_Format.Text;
string press = text6_Press.Text;
string ar = text8_ar.Text;
string kommentar = text7_Kommentar.Text;

try
{
    noupdrecs = db.UpdateRecord(row, artist_no,
album,
    format, press, ar, kommentar);
}
finally
{ }

if (noupdrecs > 0)
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces(sida);

    DateTime Now = DateTime.Now;

    string add = "<h1>Post i databas har

uppdaterats</h1>";

    add += "<h2>Data:</h2>";
    add += "<h2><div id=visakommentar>Artist_no = "
+ artist_no + "<br />";

    add += "Skivnummer = " + row + "<br />";
    add += "Album = " + text4_Album.Text + "<br

/>";

    add += "Format = " + text5_Format.Text + "<br

/>";

    add += "Press = " + text6_Press.Text + "<br

/>";

    add += "Ar = " + text8_ar.Text + "<br />";
    add += "Kommentar (se nedan): </div></h2>";
    add += "<div id=visakommentar>" + "<div>" +
text7_Kommentar.Text + "</div>";

    add += "<h3>" + "Sidan skapades: " + Now +

"</h3>";

    display.InnerHtml = add;
}
else
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces(sida);

    display.InnerHtml = "<h2>Inga erhållna värden

från databas.</h2>";
}
}
}

```

}

}

}

Klass för att visa hela innehållet i Skivtabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class Visa_databas
    {
        Clear clr = new Clear();

        //Hämtar hela databasen
        public void button_visa_databas(GridView gridArtister, Page sida)
        {
            if (gridArtister.Visible == false)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces(sida);

                gridArtister.Visible = true;
            }
        }
    }
}
```

Klass som visar kommentarpanelen för Skivtabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc;
using TF.Namespace.Controls;
using Common_Tasks;

namespace DB_Callcode.Skivor
{
    public class Visa_KommentarPanel
    {
        Clear clr = new Clear();
        Proc_act db = new Proc_act();

        //Visar panel för inmatning av en artist kort-biografi:
        public void button_kommentarpanel(PnlKommentar, Page sida)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces(sida);

            pnlKommentar.Visible = true;
        }
    }
}
```

Mellanskiktet - Komponent: DB_Callcode_Artist

Olika klasser med funktionaliteter för Artisttabellen

Klass som raderar innehållet i alla textrutor och dropdownmenyer

```
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class Borja_om
    {
        Page sida = new Page();

        //Raderar innehållet i alla textrutor och dropdownmenyer:
        public void button_Borja_om()
        {
            Control myForm = sida.FindControl("form1");
            foreach (Control ctl in myForm.Controls)
            {
                if
                (ctl.GetType().ToString().Equals("System.Web.UI.WebControls.TextBox"))
                {
                    ((TextBox)ctl).Text = "";
                }
            }
        }
    }
}
```

Klass som hanterar kryssrutor för gridview gridArtister och grdResult samt ser till att motsvarande poster i databasen är (av)markerade.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class chk_Gridviews
    {
        SetChks_Artist db = new SetChks_Artist();

        //Kryssruta för gridview gridArtister och grdResult:
        public void chkArtval_CheckedChanged(GridView gridArtister_Artist,
        GridView grdResult_Artist)
        {
            GridView crossgrid = new GridView();

            if (grdResult_Artist.Visible == true)
            {
                crossgrid = grdResult_Artist;
            }
            else if (gridArtister_Artist.Visible == true)
            {
                crossgrid = gridArtister_Artist;
            }

            foreach (GridViewRow gvRow in crossgrid.Rows)
            {
                DerivCheckBox chkssel =
                (DerivCheckBox)gvRow.FindControl("chkArtval");

                if (chkssel.Checked == true)
                {
                    Int32 val = 1;
                    Int32 row = Int32.Parse(chkssel.UnText);

                    //Sätter markeringsvärden för markerad rad:
                    gvRow.BackColor =
                    System.Drawing.ColorTranslator.FromHtml("#738A9C");
                    gvRow.ForeColor =
                    System.Drawing.ColorTranslator.FromHtml("#F7F7F7");
                }
            }
        }
    }
}
```



```

        gvRow.Font.Bold = true;

        try
        {
            db.UpdateRecord_Artist_ChkBox(row, val);
        }
        finally
        { }

    }
    else if (chkssel.Checked == false)
    {

        Int32 val = 0;
        Int32 row = Int32.Parse(chkssel.UnText);

        //Sätter tillbaka markeringsvärden för avmarkerad rad:
        gvRow.BackColor =
System.Drawing.ColorTranslator.FromHtml("#F7F7F7");
        gvRow.ForeColor =
System.Drawing.ColorTranslator.FromHtml("#4A3C8C");
        gvRow.Font.Bold = false;

        try
        {
            db.UpdateRecord_Artist_ChkBox(row, val);
        }
        finally
        { }

    }

}

}

}

```

Klass för inmatning av data till kortbio (via kortbiopanel) i Artist-abellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class dbtn_Kortbio_Input
    {
        Clear_Artist clr = new Clear_Artist();
        DB_proc_Artist.Artist db = new DB_proc_Artist.Artist();

        //Matar in data till kortibo (via kortbiopanel) i artist-tabell
        public void dynbtnKortbio(HtmlGenericControl display,
            TextBox txtKortbio, TextBox txtArtistNo, Page sida)
        {
            int noaddkortbio = 0;

            try
            {
                noaddkortbio = db.AddKortbio(txtKortbio.Text,
Int32.Parse(txtArtistNo.Text));
            }
            finally
            { }

            if (noaddkortbio > 0)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces_Artist(sida);

                DateTime Now = DateTime.Now;

                string add = "<h1>Ny Kortbio har lagts till
databasen</h1>";
                add += "<h2>Kortbio för artistno: " + txtArtistNo.Text +
"</h2>";
                add += "<h2>Inmatat till DB:</h2> ";
                add += "<div id=kortbiopanel_input>" + txtKortbio.Text +
"</div>";

                add += "<h3>" + "Sidan skapades: " + Now + "</h3>";

                display.InnerHtml = add;
            }
        }
    }
}
```

```

    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }

}

}
}

```

Klass för hämtning av kortio från Artisttabellen (via kortbiopanelen).

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespaces.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class dbtn_Kortbio_Output
    {
        Clear_Artist clr = new Clear_Artist();
        DB_proc_Artist.Artist db = new DB_proc_Artist.Artist();

        //Hämtar kortio från artist-tabell (via kortbiopanel)
        public void dynbtnKortbioFetch(HtmlGenericControl display,
        Panel pnlKortbio, TextBox txtKortbio, TextBox txtArtistNo,
        Page sida)
        {
            int rowcount = 0;

            try
            {
                DataSet ds = new DataSet();

                ds = db.FetchKortbio(Int32.Parse(txtArtistNo.Text));
                txtKortbio.Text =
ds.Tables[0].Rows[0]["Kortbio"].ToString();
                rowcount = int.Parse(ds.Tables[0].Rows.Count.ToString());
            }
            finally
            { }

            if (rowcount == 1)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces_Artist(sida);

                pnlKortbio.Visible = true;
            }
            else
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces_Artist(sida);
            }
        }
    }
}
```

```
        display.InnerHtml = "<h3>Inga erhållna värden från  
databas.</h3>";  
    }  
}  
  
}
```

Klass för att visa en artist kortbiografi.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class dbtn_Visa_Kortbio
    {
        Clear_Artist clr = new Clear_Artist();
        DB_proc_Artist.Artist db = new DB_proc_Artist.Artist();

        //Visar en artists kortbiografi:
        public void grd_Artistdata_Artist_RowCommand
            (HtmlGenericControl display,
            GridViewCommandEventArgs e, Page sida)
        {
            if (e.CommandName == "Artist_kortbio")
            {
                string row = (string)e.CommandArgument;
                string kortbio = "";
                string artist = "";
                int rowcount = 0;

                try
                {
                    DataSet ds = new DataSet();
                    ds = db.CD_VisaKortBio(row);

                    kortbio = ds.Tables[0].Rows[0]["Kortbio"].ToString();
                    artist = ds.Tables[0].Rows[0]["Artist"].ToString();

                    rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());
                }
                finally
                { }

                if (rowcount > 0)
                {
                    //Rensar display från text och gridviews
                }
            }
        }
    }
}
```

```

        clr.Clean_surfaces_Artist(sida);

        DateTime Now = DateTime.Now;

        string add = "<h1>Kort biografi om " + artist +
":</h1>";

        add += "<div id=visakortbiodata>" + kortbio + "</div>";

        display.InnerHtml = add;

    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}

}
}

```

Klass för hämtning av en rad från Artisttabellen som skall uppdateras.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class Fetch
    {
        Clear_Artist clr = new Clear_Artist();
        DB_proc_Artist.Artist db = new DB_proc_Artist.Artist();

        //Hämtar en rad från aktuell tabell som skall uppdateras:
        public void button_fetch(GridView gridArtister_Artist,
            GridView grdResult_Artist, HtmlGenericControl display,
            HtmlGenericControl display2, Page sida, TextBox text3_Artist,
            DropDownList text9_VPfr, DropDownList text10_VPtill,
            DropDownList text11_Ursland, TextBox text12_Ursstad,
            DropDownList text13_Musiktyp, TextBox text14_Kortbio)
        {

            GridView fetchgrid = new GridView();

            int a = 0;
            int rowcount = 0;

            if (grdResult_Artist.Visible == true)
            {
                fetchgrid = grdResult_Artist;
            }
            else if (gridArtister_Artist.Visible == true)
            {
                fetchgrid = gridArtister_Artist;
            }

            foreach (GridViewRow gvRow in fetchgrid.Rows)
            {

                DerivCheckBox chkssel =
                (DerivCheckBox)gvRow.FindControl("chkArtval");
```



```

        if (chkssel.Checked == true)
        {
            a++;
        }
    }

    if (a == 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(sida);

        display.InnerHtml = "<h2>Ingen kryssruta ifyllld.</h2>";
    }
    else if (a > 1)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(sida);

        display.InnerHtml = "<h2>Endast EN kryssruta får vara
ifylld.</h2>";
    }
    else
    {
        try
        {
            DataSet ds = new DataSet();
            ds = db.FetchRecord();

            grdResult_Artist.DataSource = ds;
            grdResult_Artist.DataBind();

            text3_Artist.Text =
ds.Tables[0].Rows[0]["Artist"].ToString();
            text9_VPfr.Text =
ds.Tables[0].Rows[0]["VPfr"].ToString();
            text10_VPtill.Text =
ds.Tables[0].Rows[0]["VPtill"].ToString();
            text11_Ursland.Text =
ds.Tables[0].Rows[0]["Ursland"].ToString();
            text12_Ursstad.Text =
ds.Tables[0].Rows[0]["Ursstad"].ToString();
            text13_Musiktyp.Text =
ds.Tables[0].Rows[0]["Musiktyp"].ToString();
            text14_Kortbio.Text =
ds.Tables[0].Rows[0]["Kortbio"].ToString();

            rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());
        }
        finally
        { }

        if (rowcount == 1)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(sida);

```

```

grdResult_Artist.HeaderRow.FindControl("chkArtval_ALL").Visible = false;

        grdResult_Artist.Visible = true;

        DateTime Now = DateTime.Now;

        string add1 = "<h1>Post har hämtats till databas för
uppdatering</h1>";
        string add2 = "<h3>" + "Sidan skapades: " + Now +
"</h3>";
        display.InnerHtml = add1;
        display2.InnerHtml = add2;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}

}

}

}

```

Klass för hantering av headerkryssruta för gridview grdResult.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class hchk_grdResult
    {
        SetChks_Artist db = new SetChks_Artist();

        //Headerkryssruta för gridview grdResult:
        public void chkArtvalALLGrdRes_CheckedChanged(Gridview
grdResult_Artist,
        object sender)
        {
            foreach (GridViewRow gvRow in grdResult_Artist.Rows)
            {
                DerivCheckBox chkssel =
(DerivCheckBox) (gvRow.Cells[0].FindControl("chkArtval"));
                chkssel.Checked = ((DerivCheckBox) sender).Checked;

                if (chkssel.Checked == true)
                {
                    Int32 val = 1;
                    Int32 row = Int32.Parse(chkssel.UnText);

                    try
                    {
                        db.UpdateRecord_Artist_ChkBox(row, val);
                    }
                    finally
                    {
                        { }
                    }
                }
                else if (chkssel.Checked == false)
                {
                    Int32 val = 0;
                    Int32 row = Int32.Parse(chkssel.UnText);

                    try
                    {
                        db.UpdateRecord_Artist_ChkBox(row, val);
                    }
                }
            }
        }
    }
}
```

```
        finally  
        {  
        }  
    }  
}  
}
```

Klass för hantering av headerkryssruta för gridview gridArtister.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class hchk_gridArtister
    {
        SetChks_Artist db = new SetChks_Artist();

        //Headerkryssruta för gridview gridArtister:
        public void chkArtvalALLMain_CheckedChanged(Gridview
gridArtister_Artist, object sender)
        {
            foreach (GridViewRow gvRow in gridArtister_Artist.Rows)
            {
                DerivCheckBox chkssel =
(DerivCheckBox) (gvRow.Cells[0].FindControl("chkArtval"));
                chkssel.Checked = ((DerivCheckBox) sender).Checked;

                if (chkssel.Checked == true)
                {
                    Int32 val = 1;

                    try
                    {
                        db.UpdateRecord_Artist_ChkBox(val);
                    }
                    finally
                    { }
                }
                else if (chkssel.Checked == false)
                {
                    Int32 val = 0;

                    try
                    {
                        db.UpdateRecord_Artist_ChkBox(val);
                    }
                    finally

```

```

    { }
  }
}

```

Klass för tillägg av post i Artisttabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Collections;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class Lagg_till
    {
        Clear_Artist clr = new Clear_Artist();
        DB_proc_Artist.Artist db = new DB_proc_Artist.Artist();

        //Lägger till data i databas
        public void button_lagg_till(HtmlGenericControl display,
            TextBox text3_Artist, DropDownList text9_VPfr,
            DropDownList text10_VPtill, DropDownList text11_Ursland,
            TextBox text12_Ursstad, DropDownList text13_Musiktyp,
            TextBox text14_Kortbio, BaseValidator vldArtistNotEmpty,
            BaseValidator vldArtistOnlyLetters, Page sida)
        {
            Int32 val = 0;
            Int32 noaddrecs = 0;
            string errorsum = "";
            int rowcount = 0;

            if (!sida.IsValid)
            {
                Dictionary<BaseValidator, string> dict = new
                Dictionary<BaseValidator, string>();
                dict.Add(vldArtistNotEmpty, "<h2>Ej matat in artistnamn.
                Försök igen.</h2>");
                dict.Add(vldArtistOnlyLetters, "<h2>Endast bokstäver är
                tillåtna vid angivelse av ursprungsstad.</h2>");

                rowcount = dict.Count;

                //Value- och key-kollektioner initieras mha Interface:
                ICollection keyKollektion = dict.Keys;
                ICollection valueKollektion = dict.Values;

                //Skapar arrays och kopierar kollektionerna till desamma:
                BaseValidator[] keys = new BaseValidator[rowcount];
                string[] values = new string[rowcount];
            }
        }
    }
}
```

```

keyKollektion.CopyTo(keys, 0);
valueKollektion.CopyTo(values, 0);

//Utvärderar fält mha validationskontroller och skriver
//felmeddelande till gemensam sträng:

for (int i = 0; i < rowcount; i++)
{
    if (!keys[i].IsValid)
    {
        errorsum += values[i];
    }
}

//Rensar display från text och gridviews
clr.Clean_surfaces_Artist(sida);

//Skriver gemensam felmeddelande-sträng till display:
display.InnerHtml = errorsum;
}
else
{
    try
    {
        noaddrescs = db.AddRecord(val, text3_Artist.Text,
            text9_VPfr.Text, text10_VPtill.Text,
text11_Ursland.Text, text12_Ursstad.Text,
            text13_Musiktyp.Text, text14_Kortbio.Text);
    }
    finally
    { }

    if (noaddrescs > 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(sida);

        DateTime Now = DateTime.Now;

        string add = "<h1>Ny post har lagts till
databasen</h1>";

        add += "<h2>Data:</h2>";
        add += "<h2><div id=visakommentar>KryssruteVärde = " +
val + "<br />";

        add += "Artist = " + text3_Artist.Text + "<br />";
        add += "VPfr = " + text9_VPfr.Text + "<br />";
        add += "VPtill = " + text10_VPtill.Text + "<br />";
        add += "Ursland = " + text11_Ursland.Text + "<br />";
        add += "Ursstad = " + text12_Ursstad.Text + "<br />";
        add += "Musiktyp = " + text13_Musiktyp.Text + "<br />";
        add += "Kortbio (se nedan): </div></h2>" + "<div
id=visakortbiodata>";
        add += text14_Kortbio.Text + "</div>";

        add += "<h3>" + "Sidan skapades: " + Now + "</h3>";

        display.InnerHtml = add;
    }
    else

```



```

        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(sida);

            display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
        }
    }

}

}

```

Klass som fungerar som en omstart och rensar alla inmatningsrutor odyl + display.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class Refresh_TB
    {
        //Fungerar som en omstart och rensar alla inmatningsrutor odyl +
display:
        public void button_Refresh_TB(HttpServerUtility server, string
retursida)
        {
            SetChks_Artist db = new SetChks_Artist();
            Int32 val = 0;

            try
            {
                db.UpdateRecord_Artist_ChkBox(val);
                server.Transfer(retursida);
            }
            finally
            {
            }
        }
    }
}
```

Klass som söker genom Artisttabellen enligt givna kriterier.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class Sok
    {
        Clear_Artist clr = new Clear_Artist();
        DB_proc_Artist.Artist db = new DB_proc_Artist.Artist();

        DataSet ds = new DataSet();

        //Tar emot angivelsen av hur lång tid cache:n skall verka
        private double delay;
        public double Delay
        { set { delay = value; }}

        //Söker i databas:
        public void button_sok(GridView grdResult_Artist, Page sida,
            HtmlGenericControl display, TextBox text3_Artist,
            DropDownList text9_VPfr, DropDownList text10_VPtill,
            DropDownList text11_Ursland, TextBox text12_Ursstad,
            DropDownList text13_Musiktyp, TextBox text14_Kortbio)
        {
            int rowcount = 0;

            string artist = text3_Artist.Text;
            string vpfr = text9_VPfr.Text;
            string vptill = text10_VPtill.Text;
            string ursland = text11_Ursland.Text;
            string ursstad = text12_Ursstad.Text;
            string musiktyp = text13_Musiktyp.Text;
            string kortbio = text14_Kortbio.Text;

            string res = "";
            string cnt = "";

            //Använder alla sökkriterier för att generera unika
            cachenycklar:
            ds = (DataSet)HttpRuntime.Cache["cache" + artist + vpfr +
                vptill + ursland
                    + ursstad + musiktyp + kortbio];
```

```

if (ds == null)
{
    try
    {
        ds = db.SearchRecords(artist, vpfr, vptill, ursland
            , ursstad, musiktyp, kortbio);

        //Cache för varje unik sökning + 20 sek timeout (värdet
av delay):
        HttpRuntime.Cache.Insert("cache" + artist + vpfr +
vptill + ursland
        + ursstad + musiktyp + kortbio, ds, null,
DateTime.Now.AddSeconds(delay),
        TimeSpan.Zero);

        cnt = HttpRuntime.Cache.Count.ToString();
        res = "Skapad och adderad till cache.";
    }
    finally
    { }
}
else
{
    cnt = HttpRuntime.Cache.Count.ToString();
    res = "Hämtad från cache.";
}

grdResult_Artist.DataSource = ds;
grdResult_Artist.DataBind();

//Kollar så dataset har innehåll + exekvering är ok:
rowcount = ds.Tables[0].Rows.Count;

if (rowcount > 0)
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces_Artist(sida);

    grdResult_Artist.Visible = true;

    DateTime Now = DateTime.Now;

    string add = "<h1>" + rowcount + " Post(er)
hittades:</h1>";

    add += "<h2>" + "Cache status: " + res + "<br />";
    add += "Antal objekt i cache: " + cnt + "<br />";
    add += "Delay status: " + delay + "</h2>";

    add += "<h3>" + "Sidan skapades: " + Now + "</h3>";

    display.InnerHtml = add;
}
else
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces_Artist(sida);
}

```

```
        display.InnerHtml = "<h2>Inga erhållna värden från  
databas.</h2>";  
    }  
}  
  
}  
  
}
```

Klass för att ta bort post(er) i Artisttabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class Ta_bort
    {
        Clear_Artist clr = new Clear_Artist();

        //Tar bort data från databas
        public void button_ta_bort(GridView gridArtister_Artist,
            GridView grdResult_Artist, HtmlGenericControl display,
            Page sida)
        {
            DB_proc_Artist.Artist db = new DB_proc_Artist.Artist();

            Int32 val = 1;
            Int32 nodelrecs = 0;
            Int32 a = 0;

            GridView delgrid = new GridView();

            if (grdResult_Artist.Visible == true)
            {
                delgrid = grdResult_Artist;
            }
            else if (gridArtister_Artist.Visible == true)
            {
                delgrid = gridArtister_Artist;
            }

            foreach (GridViewRow gvRow in delgrid.Rows)
            {
                DerivCheckBox chkssel =
                (DerivCheckBox)gvRow.FindControl("chkArtval");

                if (chkssel.Checked == true)
                {

```

```

        a++;
    }
}

if (a == 0)
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces_Artist(sida);

    string mess = "<h2>Ej kryssat för någon ruta. ";
    mess += "Inget kan därmed tas bort. Kryssa för minst en
ruta</h2>";

    display.InnerHtml = mess;
}
else if (a >= 1)
{

    try
    {
        nodelrecs = db.DelRecord(val);
    }
    finally
    { }

    if (nodelrecs > 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(sida);

        DateTime Now = DateTime.Now;

        string add = "<h1>" + nodelrecs + " Post(er) har tagits
bort från databasen</h1>";
        add += "<h3>" + "Sidan skapades: " + Now + "</h3>";
        display.InnerHtml = add;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Artist(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}

}

}

}

```

Klass som uppdaterar Artisttabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class Uppdatera
    {
        Clear_Artist clr = new Clear_Artist();

        //Uppdaterar databas:
        public void button_oppdatera(GridView gridArtister_Artist,
            GridView grdResult_Artist, HtmlGenericControl display,
            Page sida, TextBox text3_Artist, DropDownList text9_VPfr,
            DropDownList text10_VPtill, DropDownList text11_Ursland,
            TextBox text12_Ursstad, DropDownList text13_Musiktyp,
            TextBox text14_Kortbio)
        {
            Int32 noupdrecs = 0;
            GridView updategrid = new GridView();

            if (grdResult_Artist.Visible == true)
            {
                updategrid = grdResult_Artist;
            }
            else if (gridArtister_Artist.Visible == true)
            {
                updategrid = gridArtister_Artist;
            }

            foreach (GridViewRow gvRow in updategrid.Rows)
            {
                DerivCheckBox chkssel =
                (DerivCheckBox)gvRow.FindControl("chkArtval");

                if (chkssel.Checked == true)
                {
                    DB_proc_Artist.Artist db = new
                    DB_proc_Artist.Artist();
                }
            }
        }
    }
}
```



```

Int32 row = Int32.Parse(chksel.UnText);
string artist = text3_Artist.Text;
string vpfr = text9_VPfr.Text;
string vptill = text10_VPtill.Text;
string ursland = text11_Ursland.Text;
string ursstad = text12_Ursstad.Text;
string musiktyp = text13_Musiktyp.Text;
string kortbio = text14_Kortbio.Text;

try
{
    noupdrecs = db.UpdateRecord(row, artist, vpfr,
        vptill, ursland, ursstad, musiktyp, kortbio);
}
finally
{ }

if (noupdrecs > 0)
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces_Artist(sida);

    DateTime Now = DateTime.Now;

    string add = "<h1>Post i databas har

uppdaterats</h1>";

    add += "<h2>Data:</h2>";
    add += "<h2><div id=visakommentar>Artist_no = "

    add += "Artist = " + text3_Artist.Text + "<br

    add += "VPfr = " + text9_VPfr.Text + "<br />";
    add += "VPtill = " + text10_VPtill.Text + "<br

    add += "Ursland = " + text11_Ursland.Text +

    add += "Ursstad = " + text12_Ursstad.Text +

    add += "Musiktyp = " + text13_Musiktyp.Text +

    add += "Kortbio (se nedan): </div></h2>" +

    add += text14_Kortbio.Text + "</div>";

    add += "<h3>" + "Sidan skapades: " + Now +

    "</h3>";

    display.InnerHtml = add;
}
else
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces_Artist(sida);

    display.InnerHtml = "<h2>Inga erhållna värden

från databas.</h2>";
}

```

}

}

}

}

}

Klass för att visa hela innehållet i Artisttabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class Visa_databas
    {
        Clear_Artist clr = new Clear_Artist();

        //Hämtar hela databasen
        public void button_visa_databas(GridView gridArtister_Artist, Page
sida)
        {
            if (gridArtister_Artist.Visible == false)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces_Artist(sida);

                gridArtister_Artist.Visible = true;
            }
        }
    }
}
```

Klass som visar kortbiopanelen för Artisttabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Artist;
using TF.Namespace.Controls;
using Common_Tasks_Artist;

namespace DB_Callcode_Artist.Artist
{
    public class Visa_KortBioPanel
    {
        Clear_Artist clr = new Clear_Artist();
        DB_proc_Artist.Artist db = new DB_proc_Artist.Artist();

        //Visar panel för inmatning av en artist kort-biografi:
        public void button_kortbiopanel(PnlKortbio, Page sida)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Artist(sida);

            pnlKortbio.Visible = true;
        }
    }
}
```

Mellanskiktet - Komponent: DB_Callcode_Inkop

Olika klasser med funktionaliteter för Inkopstabellen

Klass som raderar innehållet i alla textrutor och dropdownmenyer.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class Borja_om
    {
        Page sida = new Page();

        //Raderar innehållet i alla textrutor och dropdownmenyer:
        public void button_Borja_om()
        {
            Control myForm = sida.FindControl("Form1");
            foreach (Control ctl in myForm.Controls)
                if
(ctl.GetType().ToString().Equals("System.Web.UI.WebControls.TextBox"))
                    ((TextBox)ctl).Text = "";
        }
    }
}
```

Klass som hanterar kryssrutor för gridview gridArtister och grdResult samt ser till att motsvarande poster i databasen är (av)markerade.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class chk_Gridviews
    {
        SetChks_Inkop db = new SetChks_Inkop();

        //Kryssruta för gridview gridArtister och grdResult:
        public void chkInkval_CheckedChanged(GridView gridArtister_Inkop,
        GridView grdResult_Inkop)
        {
            GridView crossgrid = new GridView();

            if (grdResult_Inkop.Visible == true)
            {
                crossgrid = grdResult_Inkop;
            }
            else if (gridArtister_Inkop.Visible == true)
            {
                crossgrid = gridArtister_Inkop;
            }

            foreach (GridViewRow gvRow in crossgrid.Rows)
            {
                DerivCheckBox chksel =
                (DerivCheckBox)gvRow.FindControl("chkInkval");

                if (chksel.Checked == true)
                {
                    Int32 val = 1;
                    Int32 row = Int32.Parse(chksel.UnText);

                    //Sätter markeringsvärden för markerad rad:
                    gvRow.BackColor =
                    System.Drawing.ColorTranslator.FromHtml("#738A9C");
                }
            }
        }
    }
}
```

```

        gvRow.ForeColor =
System.Drawing.ColorTranslator.FromHtml("#F7F7F7");
        gvRow.Font.Bold = true;

        try
        {
            db.UpdateRecord_ChkBox(row, val);
        }
        finally
        { }

    }
    else if (chkssel.Checked == false)
    {

        Int32 val = 0;
        Int32 row = Int32.Parse(chkssel.UnText);

        //Sätter tillbaka markeringsvärden för avmarkerad rad:
        gvRow.BackColor =
System.Drawing.ColorTranslator.FromHtml("#F7F7F7");
        gvRow.ForeColor =
System.Drawing.ColorTranslator.FromHtml("#4A3C8C");
        gvRow.Font.Bold = false;

        try
        {
            db.UpdateRecord_ChkBox(row, val);
        }
        finally
        { }

    }

}

}

}

}

```


Klass som hanterar de dynamiska knapparna för visning av kommentarer respektive artistdata.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;
namespace DB_Callcode_Inkop.Inkop
{
    public class dbtn_KomArt
    {
        Clear_Inkop clr = new Clear_Inkop();
        DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

        //Kommentarknapp och Artistknapp:
        public void gridArtister_Inkop_RowCommand(
            GridView grd_Artistdata_Inkop, HtmlGenericControl display,
            HtmlGenericControl display2, Page sida, GridViewCommandEventArgs e)
        {

            if (e.CommandName == "Kommentar")
            {

                int row = Convert.ToInt32(e.CommandArgument);
                string kom = "";
                int rowcount = 0;

                try
                {
                    DataSet ds = new DataSet();
                    ds = db.CD_Kommentarer(row);
                    kom = ds.Tables[0].Rows[0]["Komt"].ToString();
                    rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());
                }
                finally
                { }

                if (rowcount > 0)
                {
                    //Rensar display från text och gridviews
                    clr.Clean_surfaces_Inkop(sida);

                    DateTime Now = DateTime.Now;

                    string add = "<h1>Kommentar No: " + row + "</h1>";
                    add += "<div id=visakommentar>" + kom + "</div>";
                }
            }
        }
    }
}
```

```

        add += "<h3>" + "Sidan skapades: " + Now + "</h3>";
        display.InnerHtml = add;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }

}
else if (e.CommandName == "Artist")
{

    int row = Convert.ToInt32(e.CommandArgument);
    int rowcount = 0;

    try
    {
        DataSet ds = new DataSet();
        ds = db.CD_VisaArtistData(row);

        grd_Artistdata_Inkop.DataSource = ds;
        grd_Artistdata_Inkop.DataBind();

        rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());

    }
    finally
    { }

    if (rowcount > 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        grd_Artistdata_Inkop.Visible = true;

        DateTime Now = DateTime.Now;

        string add1 = "<h1>Artistdata No: " + row + "</h1>";
        string add2 = "<h3>" + "Sidan skapades: " + Now +
"</h3>";

        display.InnerHtml = add1;
        display2.InnerHtml = add2;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}
}

```

}

}

}

Klass för inmatning av längre kommentarer till Inkopstabell via kommentarpanel.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class dbtn_Kommentar_Input
    {
        Clear_Inkop clr = new Clear_Inkop();
        DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

        //Matar in längre kommentar till skivtabell:
        public void dynbtnKommentar(HtmlGenericControl display,
            TextBox txtKommentar, TextBox txtKommentarNo, Page sida)
        {
            int noaddkommentar = 0;

            try
            {
                noaddkommentar = db.AddKommentar(txtKommentar.Text,
                Int32.Parse(txtKommentarNo.Text));
            }
            finally
            { }

            if (noaddkommentar > 0)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces_Inkop(sida);

                DateTime Now = DateTime.Now;

                string add = "<h1>Ny kommentar har lagts till
databasen</h1>";
                add += "<h2>Kommentar för artistno: " + txtKommentarNo.Text
                + "</h2>";
                add += "<h2>Inmatat till DB:</h2> ";
                add += "<div id=kommentarpanel_input>" + txtKommentar.Text
                + "</div>";
            }
        }
    }
}
```

```

        add += "<h3>" + "Sidan skapades: " + Now + "</h3>";

        display.InnerHtml = add;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }

}

}

}

```

Klass för hämtning av längre kommentar från Inkopstabell (via kommentarpanel).

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class dbtn_Kommentar_Output
    {
        Clear_Inkop clr = new Clear_Inkop();
        DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

        //Hämtar längre kommentar från skivtabell (via kommentarpanel)
        public void dynbtnKommentarFetch(HtmlGenericControl display,
        Panel pnlKommentar_Inkop, TextBox txtKommentar,
        TextBox txtKommentarNo, Page sida)
        {
            int rowcount = 0;

            try
            {
                DataSet ds = new DataSet();

                ds = db.FetchKommentar(Int32.Parse(txtKommentarNo.Text));
                txtKommentar.Text =
ds.Tables[0].Rows[0]["Komt"].ToString();
                rowcount = int.Parse(ds.Tables[0].Rows.Count.ToString());
            }
            finally
            { }

            if (rowcount == 1)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces_Inkop(sida);

                pnlKommentar_Inkop.Visible = true;
            }
            else
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces_Inkop(sida);
            }
        }
    }
}
```

```
        display.InnerHtml = "<h3>Inga erhållna värden från  
databas.</h3>";  
    }  
}  
  
}  
  
}
```

Klass för att visa en artist kortbiografi.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{

    public class dbtn_Visa_Kortbio
    {
        Clear_Inkop clr = new Clear_Inkop();
        DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

        //Visar en artists kortbiografi:
        public void grd_Artistdata_Inkop_RowCommand(HtmlGenericControl
display,
        GridViewCommandEventArgs e, Page sida)
        {
            if (e.CommandName == "Artist_kortbio")
            {

                string row = (string)e.CommandArgument;
                string kortbio = "";
                string artist = "";
                int rowcount = 0;

                try
                {
                    DataSet ds = new DataSet();
                    ds = db.CD_VisaKortBio(row);

                    kortbio = ds.Tables[0].Rows[0]["Kortbio"].ToString();
                    artist = ds.Tables[0].Rows[0]["Artist"].ToString();

                    rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());

                }
                finally
                { }

                if (rowcount > 0)
                {
```



```

//Rensar display från text och gridviews
clr.Clean_surfaces_Inkop(sida);

DateTime Now = DateTime.Now;

string add = "<h1>Kort biografi om " + artist +
":</h1>";

add += "<div id=visakortbiodata>" + kortbio + "</div>";

display.InnerHtml = add;

}
else
{
//Rensar display från text och gridviews
clr.Clean_surfaces_Inkop(sida);

display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
}
}

}

}

```

Klass för hämtning av en rad från Inkopstabellen som skall uppdateras

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class Fetch
    {
        Clear_Inkop clr = new Clear_Inkop();
        DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

        //Hämtar en rad från aktuell tabell som skall uppdateras:
        public void button_fetch(GridView gridArtister_Inkop, GridView
grdResult_Inkop,
        HtmlGenericControl display, HtmlGenericControl display2,
        Page sida, TextBox text3_Artist, TextBox text4_Album,
        DropDownList text5_Format, DropDownList text6_Press, DropDownList
text8_ar,
        TextBox text7_Kommentar, TextBox text15_Inm_dat, DropDownList
text16_Kop_grad,
        DropDownList text17_Kop_kat, TextBox text18_Ca_pris)
        {
            DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

            GridView fetchgrid = new GridView();

            int a = 0;
            int rowcount = 0;

            if (grdResult_Inkop.Visible == true)
            {
                fetchgrid = grdResult_Inkop;
            }
            else if (gridArtister_Inkop.Visible == true)
            {
                fetchgrid = gridArtister_Inkop;
            }

            foreach (GridViewRow gvRow in fetchgrid.Rows)
            {
```

```

        DerivCheckBox chkssel =
(DerivCheckBox) gvRow.FindControl("chkInkval");

        if (chkssel.Checked == true)
        {
            a++;
        }
    }

    if (a == 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        display.InnerHtml = "<h2>Ingen kryssruta ifylld.</h2>";
    }
    else if (a > 1)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        display.InnerHtml = "<h2>Endast EN kryssruta får vara
ifylld.</h2>";
    }
    else
    {
        try
        {
            DataSet ds = new DataSet();
            ds = db.FetchRecord();

            grdResult_Inkop.DataSource = ds;
            grdResult_Inkop.DataBind();

            text3_Artist.Text =
ds.Tables[0].Rows[0]["Art_no"].ToString();
            text4_Album.Text =
ds.Tables[0].Rows[0]["Titel"].ToString();
            text5_Format.Text =
ds.Tables[0].Rows[0]["Form"].ToString();
            text6_Press.Text =
ds.Tables[0].Rows[0]["Land"].ToString();
            text8_ar.Text = ds.Tables[0].Rows[0]["Utg"].ToString();
            text7_Kommentar.Text =
ds.Tables[0].Rows[0]["Komt"].ToString();
            text15_Inm_dat.Text =
ds.Tables[0].Rows[0]["Inm_dat"].ToString();
            text16_Kop_grad.Text =
ds.Tables[0].Rows[0]["Kop_grad"].ToString();
            text17_Kop_kat.Text =
ds.Tables[0].Rows[0]["Kop_kat"].ToString();
            text18_Ca_pris.Text =
ds.Tables[0].Rows[0]["Ca_pris"].ToString();

            rowcount =
int.Parse(ds.Tables[0].Rows.Count.ToString());
        }
        finally
        { }
    }
}

```

```

        if (rowcount == 1)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(sida);

grdResult_Inkop.HeaderRow.FindControl("chkInkval_ALL").Visible = false;

            grdResult_Inkop.Visible = true;

            DateTime Now = DateTime.Now;

            string add1 = "<h1>Post har hämtats till databas för
uppdatering</h1>";
            string add2 = "<h3>" + "Sidan skapades: " + Now +
"</h3>";

            display.InnerHtml = add1;
            display2.InnerHtml = add2;
        }
        else
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(sida);

            display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
        }
    }

}

}

}

```

Klass för hantering av headerkryssruta för gridview grdResult.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class hchk_grdResult
    {
        SetChks_Inkop db = new SetChks_Inkop();

        //Headerkryssruta för gridview grdResult:
        public void chkInkvalALLGrdRes_CheckedChanged(GridView
grdResult_Inkop,
        object sender)
        {
            foreach (GridViewRow gvRow in grdResult_Inkop.Rows)
            {
                DerivCheckBox chkssel =
(DerivCheckBox) (gvRow.Cells[0].FindControl("chkInkval"));
                chkssel.Checked = ((DerivCheckBox) sender).Checked;

                if (chkssel.Checked == true)
                {
                    Int32 val = 1;
                    Int32 row = Int32.Parse(chkssel.UnText);

                    try
                    {
                        db.UpdateRecord_ChkBox(row, val);
                    }
                    finally
                    { }
                }

                else if (chkssel.Checked == false)
                {
                    Int32 val = 0;
                    Int32 row = Int32.Parse(chkssel.UnText);

                    try
                    {
                        db.UpdateRecord_ChkBox(row, val);
                    }
                }
            }
        }
    }
}
```

```
        finally
        { }
    }
}

}
```

Klass för hantering av headerkryssruta för gridview gridArtister.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class hchk_gridArtister
    {
        SetChks_Inkop db = new SetChks_Inkop();

        //Headerkryssruta för gridview gridArtister:
        public void chkInkvalALLMain_CheckedChanged(GridView
            gridArtister_Inkop, object sender)
        {
            foreach (GridViewRow gvRow in gridArtister_Inkop.Rows)
            {
                DerivCheckBox chkssel =
                    (DerivCheckBox) (gvRow.Cells[0].FindControl("chkInkval"));
                chkssel.Checked = ((DerivCheckBox) sender).Checked;

                if (chkssel.Checked == true)
                {
                    Int32 val = 1;

                    try
                    {
                        db.UpdateRecord_ChkBox(val);
                    }
                    finally
                    { }
                }
                else if (chkssel.Checked == false)
                {
                    Int32 val = 0;

                    try
                    {
                        db.UpdateRecord_ChkBox(val);
                    }
                    finally
                    { }
                }
            }
        }
    }
}
```

}
}
}
}

}

Klass för tillägg av post i Inkopstabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;
using System.Collections;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class Lagg_till
    {
        Clear_Inkop clr = new Clear_Inkop();
        DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

        //Lägger till data i databas
        public void button_lagg_till(HtmlGenericControl display, TextBox
text3_Artist,
        TextBox text4_Album, DropDownList text5_Format, DropDownList
text6_Press,
        DropDownList text8_ar, TextBox text7_Kommentar, TextBox
text15_Inm_dat,
        DropDownList text16_Kop_grad, DropDownList text17_Kop_kat, TextBox
text18_Ca_pris,
        BaseValidator vldDatum, BaseValidator vldCaPris, BaseValidator
vldArtistOnlyNum,
        BaseValidator vldArtistNotEmpty, BaseValidator vldArtistValidRange,
Page sida)
        {
            Int32 val = 0;
            Int32 noaddrescs = 0;
            string errorsum = "";
            int rowcount = 0;

            string vldArtistValidRangeErrText = "<h2>Artistnummer utanför
registrerade värden ";
            vldArtistValidRangeErrText += "i databas. Kolla i
artisttabellen och försök igen.</h2>";

            if (!sida.IsValid)
            {
                Dictionary<BaseValidator, string> dict = new
Dictionary<BaseValidator, string>();
                dict.Add(vldArtistNotEmpty, "<h2>Ej matat in artistnummer.
Kolla i artisttabellen och försök igen.</h2>");
                dict.Add(vldArtistOnlyNum, "<h2>Endast siffror är tillåtna
som artistnummer.</h2>");
                dict.Add(vldDatum, "<h2>Ej tillåtet datumformat
angivet.</h2>");
```

```

dict.Add(vldCaPris, "<h2>Ej tillåtet format för cirkapris
angivet.</h2>");
dict.Add(vldArtistValidRange, vldArtistValidRangeErrText);

rowcount = dict.Count;

//Value- och key-kollektioner initieras mha Interface:
ICollection keyKollektion = dict.Keys;
ICollection valueKollektion = dict.Values;

//Skapar arrays och kopierar kollektionerna till desamma:
BaseValidator[] keys = new BaseValidator[rowcount];
string[] values = new string[rowcount];
keyKollektion.CopyTo(keys, 0);
valueKollektion.CopyTo(values, 0);

//Utvärderar fält mha validationskontroller och skriver
//felmeddelande till gemensam sträng:
for (int i = 0; i < rowcount-1; i++)
{
    if (!keys[i].IsValid)
    {
        errorsum += values[i];
    }
}

if (!keys[rowcount-1].IsValid && keys[1].IsValid)
    errorsum += values[rowcount-1];

//Rensar display från text och gridviews
clr.Clean_surfaces_Inkop(sida);

//Skriver gemensam felmeddelande-sträng till display:
display.InnerHtml = errorsum;
}
else
{
    try
    {
        noaddrescs = db.AddRecord(val,
Int32.Parse(text3_Artist.Text),
        text4_Album.Text, text5_Format.Text, text6_Press.Text,
text8_ar.Text,
        text7_Kommentar.Text, text15_Inm_dat.Text,
text16_Kop_grad.Text,
        text17_Kop_kat.Text, text18_Ca_pris.Text);
    }
    finally
    { }

    if (noaddrescs > 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        DateTime Now = DateTime.Now;

```

```

databasen</h1>";

val + "<br />";

" <br />";

/>";

+ "</div>";

string add = "<h1>Ny post har lagts till

add += "<h2>Data:</h2>";
add += "<h2><div id=visakommentar>KryssruteVärde = " +

add += "Artist_no = " + text3_Artist.Text + "<br />";
add += "Album = " + text4_Album.Text + "<br />";
add += "Format = " + text5_Format.Text + "<br />";
add += "Press = " + text6_Press.Text + "<br />";
add += "År = " + text8_ar.Text + "<br />";
add += "Inmatningsdatum = " + text15_Inm_dat.Text +

add += "Köpgrad = " + text16_Kop_grad.Text + "<br />";
add += "Köpkategori = " + text17_Kop_kat.Text + "<br

add += "Cirkapris = " + text18_Ca_pris.Text + "<br />";
add += "Kommentar (se nedan): </div></h2>";
add += "<div id=visakommentar>" + text7_Kommentar.Text

add += "<h3>" + "Sidan skapades: " + Now + "</h3>";

display.InnerHtml = add;

}
else
{
    //Rensar display från text och gridviews
    clr.Clean_surfaces_Inkop(sida);

    display.InnerHtml = "<h2>Inga erhållna värden från

databas.</h2>";
}
}
}

}

}

```

Klass som fungerar som en omstart och rensar alla inmatningsrutor odyl + display.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class Refresh_TB
    {
        //Fungerar som en omstart och rensar alla inmatningsrutor odyl +
display:
        public void button_Refresh_TB(HttpServerUtility server, string
retursida)
        {

            SetChks_Inkop db = new SetChks_Inkop();
            Int32 val = 0;

            try
            {
                db.UpdateRecord_ChkBox(val);
                server.Transfer(retursida);
            }
            finally
            { }

        }

    }
}
```

Klass som söker genom Inkopstabellen enligt givna kriterier.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class Sok
    {
        Clear_Inkop clr = new Clear_Inkop();
        DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

        DataSet ds = new DataSet();

        //Tar emot angivelsen av hur lång tid cache:n skall verka
        private double delay;
        public double Delay
        { set { delay = value; } get { return delay; }}

        //Söker i databas:
        public void button_sok(GridView grdResult_Inkop, Page sida,
            HtmlGenericControl display, TextBox text3_Artist, TextBox
text4_Album,
            DropDownList text5_Format, DropDownList text6_Press, DropDownList
text8_ar,
            TextBox text7_Kommentar, TextBox text15_Inm_dat, DropDownList
text16_Kop_grad,
            DropDownList text17_Kop_kat, TextBox text18_Ca_pris)
        {
            int rowcount = 0;

            string art_no = text3_Artist.Text;
            string album = text4_Album.Text;
            string format = text5_Format.Text;
            string press = text6_Press.Text;
            string ar = text8_ar.Text;
            string kommentar = text7_Kommentar.Text;
            string inm_dat=text15_Inm_dat.Text;
            string kop_grad=text16_Kop_grad.Text;
            string kop_kat=text17_Kop_kat.Text;
            string ca_pris = text18_Ca_pris.Text;

            string res = "";
            string cnt = "";
```

```

        //Använder alla sökkriterier för att generera unika
        cachenycklar:
        ds = (DataSet)HttpRuntime.Cache["cache" + art_no + album +
        format + press
            + ar + kommentar + inm_dat + kop_grad + kop_kat +
        ca_pris];

        if (ds == null)
        {

            try
            {
                ds = db.SearchRecords(art_no, album, format, press, ar,
                kommentar, inm_dat, kop_grad, kop_kat, ca_pris);

                //Cache för varje unik sökning + 20 sek timeout (värdet
                av delay):
                HttpRuntime.Cache.Insert("cache" + art_no + album +
                format + press + ar + kommentar + inm_dat
                + kop_grad + kop_kat + ca_pris, ds, null,
                DateTime.Now.AddSeconds(delay), TimeSpan.Zero);

                cnt = HttpRuntime.Cache.Count.ToString();
                res = "Skapad och adderad till cache.";
            }
            finally
            { }

        }
        else
        {
            cnt = HttpRuntime.Cache.Count.ToString();
            res = "Hämtad från cache.";
        }

        grdResult_Inkop.DataSource = ds;
        grdResult_Inkop.DataBind();

        //Kollar så dataset har innehåll + exekvering är ok:
        rowcount = ds.Tables[0].Rows.Count;

        if (rowcount > 0)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(sida);

            grdResult_Inkop.Visible = true;

            DateTime Now = DateTime.Now;

            string add = "<h1>" + rowcount + " Post(er)
            hittades:</h1>";

            add += "<h2>" + "Cache status: " + res + "<br />";
            add += "Antal objekt i cache: " + cnt + "<br />";
            add += "Delay status: " + delay + "</h2>";

            add += "<h3>" + "Sidan skapades: " + Now + "</h3>";

            display.InnerHtml = add;

```

```

    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
    }
}

}

```

Klass för att ta bort post(er) i Inkopstabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class Ta_bort
    {
        Clear_Inkop clr = new Clear_Inkop();

        //Tar bort data från databas
        public void button_ta_bort(GridView gridArtister_Inkop, GridView
grdResult_Inkop,
        HtmlGenericControl display, Page sida)
        {
            DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

            Int32 ValInk = 1;
            Int32 nodelrecs = 0;
            Int32 a = 0;

            GridView delgrid = new GridView();

            if (grdResult_Inkop.Visible == true)
            {
                delgrid = grdResult_Inkop;
            }
            else if (gridArtister_Inkop.Visible == true)
            {
                delgrid = gridArtister_Inkop;
            }

            foreach (GridViewRow gvRow in delgrid.Rows)
            {
                DerivCheckBox chksel =
(DerivCheckBox)gvRow.FindControl("chkInkval");
```



```

        if (chkssel.Checked == true)
        {
            a++;
        }
    }

    if (a == 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        string mess = "<h2>Ej kryssat för någon ruta. ";
        mess += "Inget kan därmed tas bort. Kryssa för minst en
ruta</h2>";

        display.InnerHtml = mess;
    }
    else if (a >= 1)
    {

        try
        {
            nodelrecs = db.DelRecord(ValInk);
        }
        finally
        { }

        if (nodelrecs > 0)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(sida);

            DateTime Now = DateTime.Now;

            string add = "<h1>" + nodelrecs + " Post(er) har tagits
bort från databasen</h1>";
            add += "<h3>" + "Sidan skapades: " + Now + "</h3>";
            display.InnerHtml = add;
        }
        else
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(sida);

            display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
        }
    }

}

}

}

```

Klass som sköter om överföring av post från Inkopstabell till Skivtabell (inköpt skiva som inte längre skall vara i Inkopstabellen).

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class Transfer
    {
        Clear_Inkop clr = new Clear_Inkop();

        //Flyttar post från Inkopstabell till Skivtabell:
        public void button_transfer(GridView gridArtister_Inkop, GridView
grdResult_Inkop,
        HtmlGenericControl display, Page sida)
        {
            DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

            Int32 ValInk = 1;
            Int32 nottransfrecs = 0;
            Int32 a = 0;

            GridView delgrid = new GridView();

            if (grdResult_Inkop.Visible == true)
            {
                delgrid = grdResult_Inkop;
            }
            else if (gridArtister_Inkop.Visible == true)
            {
                delgrid = gridArtister_Inkop;
            }

            foreach (GridViewRow gvRow in delgrid.Rows)
            {
                DerivCheckBox chkssel =
(DerivCheckBox)gvRow.FindControl("chkInkval");
```

```

        if (chkssel.Checked == true)
        {
            a++;
        }
    }

    if (a == 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        string mess = "<h2>Ej kryssat för någon ruta. ";
        mess += "Inget kan därmed flyttas. Kryssa för en
ruta</h2>";
        display.InnerHtml = mess;
    }
    else if (a > 1)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        string mess = "<h2>Kryssat i mer än en ruta. ";
        mess += "Inget kan därmed flyttas. Kryssa för endast en
ruta</h2>";
        display.InnerHtml = mess;
    }
    else if (a == 1)
    {
        try
        {
            notransfrecs = db.TransferRecord(ValInk);
        }
        finally
        { }

        if (notransfrecs == 1)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(sida);

            DateTime Now = DateTime.Now;

            string add = "<h1>" + "Post har flyttats till
Skivtabell</h1>";
            add += "<h3>" + "Sidan skapades: " + Now + "</h3>";
            display.InnerHtml = add;
        }
        else
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(sida);

            display.InnerHtml = "<h2>Inga erhållna värden från
databas.</h2>";
        }
    }
}

```

}

}

}

}

Klass som uppdaterar Inkopstabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class Uppdatera
    {
        Clear_Inkop clr = new Clear_Inkop();

        //Uppdaterar databas:
        public void button_oppdatera(GridView gridArtister_Inkop, GridView
grdResult_Inkop,
        HtmlGenericControl display, Page sida, TextBox text3_Artist,
        TextBox text4_Album,
        DropDownList text5_Format, DropDownList text6_Press, DropDownList
text8_ar,
        TextBox text7_Kommentar, TextBox text15_Inm_dat, DropDownList
text16_Kop_grad,
        DropDownList text17_Kop_kat, TextBox text18_Ca_pris)
        {
            Int32 noupdrecs = 0;
            GridView updategrid = new GridView();

            if (grdResult_Inkop.Visible == true)
            {
                updategrid = grdResult_Inkop;
            }
            else if (gridArtister_Inkop.Visible == true)
            {
                updategrid = gridArtister_Inkop;
            }

            foreach (GridViewRow gvRow in updategrid.Rows)
            {

                DerivCheckBox chkssel =
(DerivCheckBox)gvRow.FindControl("chkInkval");

                if (chkssel.Checked == true)
```

```

{
    DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

    Int32 row = Int32.Parse(chksel.UnText);
    Int32 artist_no = Int32.Parse(text3_Artist.Text);
    string album = text4_Album.Text;
    string format = text5_Format.Text;
    string press = text6_Press.Text;
    string ar = text8_ar.Text;
    string kommentar = text7_Kommentar.Text;
    string Inm_dat = text15_Inm_dat.Text;
    string Kop_grad = text16_Kop_grad.Text;
    string Kop_kat = text17_Kop_kat.Text;
    string Ca_pris = text18_Ca_pris.Text;

    try
    {
        noupdrecs = db.UpdateRecord(row, artist_no,
album,
        format, press, ar, kommentar, Inm_dat,
Kop_grad,
        Kop_kat, Ca_pris);
    }
    finally
    { }

    if (noupdrecs > 0)
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        DateTime Now = DateTime.Now;

        string add = "<h1>Post i databas har

uppdaterats</h1>";

        add += "<h2>Data:</h2>";
        add += "<h2><div id=visakommentar>Artist_no = "

+ artist_no + "<br />";

        add += "Skivnummer = " + row + "<br />";
        add += "Album = " + text4_Album.Text + "<br

/>>";

        add += "Format = " + text5_Format.Text + "<br

/>>";

        add += "Press = " + text6_Press.Text + "<br

/>>";

        add += "Ar = " + text8_ar.Text + "<br />";
        add += "Inmatningsdatum = " +

text15_Inm_dat.Text + "<br />";
        add += "Köpgrad = " + text16_Kop_grad.Text +

"<br />";

        add += "Köpkategori = " + text17_Kop_kat.Text +

"<br />";

        add += "Cirkapris = " + text18_Ca_pris.Text +

"<br />";

        add += "Kommentar (se nedan): </div></h2>";
        add += "<div id=visakommentar>" + "<div>" +

text7_Kommentar.Text + "</div>";

        add += "<h3>" + "Sidan skapades: " + Now +

"</h3>";
    }
}

```

```

        display.InnerHtml = add;
    }
    else
    {
        //Rensar display från text och gridviews
        clr.Clean_surfaces_Inkop(sida);

        display.InnerHtml = "<h2>Inga erhållna värden
från databas.</h2>";
    }
}

}

}

}

```

Klass för att visa hela innehållet i Inkopstabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;
namespace DB_Callcode_Inkop.Inkop
{
    public class Visa_databas
    {
        Clear_Inkop clr = new Clear_Inkop();

        //Hämtar hela databasen
        public void button_visa_databas(GridView gridArtister_Inkop, Page
sida)
        {
            if (gridArtister_Inkop.Visible == false)
            {
                //Rensar display från text och gridviews
                clr.Clean_surfaces_Inkop(sida);

                gridArtister_Inkop.Visible = true;
            }
        }
    }
}
```


Klass som visar kommentarpanelen för Inkopstabellen.

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Text.RegularExpressions;
using System.Web.Caching;
using System.Web.UI.HtmlControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;
using Common_Tasks_Inkop;

namespace DB_Callcode_Inkop.Inkop
{
    public class Visa_KommentarPanel
    {
        Clear_Inkop clr = new Clear_Inkop();
        DB_proc_Inkop.Inkop db = new DB_proc_Inkop.Inkop();

        //Visar panel för inmatning av en artist kort-biografi:
        public void button_kommentarpanel(Panel pnlKommentar_Inkop, Page
sida)
        {
            //Rensar display från text och gridviews
            clr.Clean_surfaces_Inkop(sida);

            pnlKommentar_Inkop.Visible = true;
        }
    }
}
```

Mellanskiktet - Komponent: DB_proc

Klass med olika funktioner gällande interaktion direkt mot databas för Skivtabellen
(börjar på nästa sida)

```

using System;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Configuration;
using System.Collections.Generic;
using System.Web.Caching;
using System.Collections;
using System.Collections.Specialized;

using MySql.Data.MySqlClient;

namespace DB_proc
{
    public class Proc_act
    {
        //Anslutning mot DB:

        //Anslutningssträng mot DB
        private string connectionString;

        //Anslutning mot DB inkl felhantering:
        public Proc_act()
        {
            if (WebConfigurationManager.ConnectionStrings["lokal"] == null)
            {
                throw new ApplicationException("ConnectionString saknas i
web.config.");
            }
            else
            {
                connectionString =
WebConfigurationManager.ConnectionStrings["lokal"].ConnectionString;
            }
        }

        //Allmänna DB-funktioner för "knapptsatsen":

        //Visa hela DB:
        public DataSet GetWholeDB()
        {
            string query = "select Val, Artist, Album, Format, Press, Ar,
Kommentar, `#`, Artist_no from cd_net, artist_net where Artist_no=No order
by Artist, Album asc";

            MySqlCommand cmd = new MySqlCommand(query);

            return FillDataSet(cmd, "whole");
        }
    }
}

```

```

//Lägga till post i DB:
public int AddRecord(int val, int artist_no, string album, string
format,
                        string press, string ar, string kommentar)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    int noaddrescs = 0;

    //Skapa kommandot
    string sql = "insert into cd_net";
    sql += "(Val, Artist_no, Album, Format, Press, Ar, Kommentar)";
    sql += " values(@Val, @Artist_no, @Album, @Format, @Press, @Ar,
@Kommentar)";
    MySqlCommand cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@Val", val);
    cmd.Parameters.AddWithValue("@Artist_no", artist_no);
    cmd.Parameters.AddWithValue("@Album", album);
    cmd.Parameters.AddWithValue("@Format", format);
    cmd.Parameters.AddWithValue("@Press", press);
    cmd.Parameters.AddWithValue("@Ar", ar);
    cmd.Parameters.AddWithValue("@Kommentar", kommentar);

    try
    {
        con.Open();
        noaddrescs = cmd.ExecuteNonQuery();
        return noaddrescs;
    }
    finally
    {
        con.Close();
    }
}

//Hämtar min- och maxgränser för rangevalidator gällande
begränsingar
//av artisttabellens omfång (antal f.n inmatade artister)
//Arbetar i samband med AddRecord-funktionen ovan:
public int[] GetAggCDNet()
{
    MySqlConnection con = new MySqlConnection(connectionString);
    int min = 0;
    int max = 0;
    int[] aggstore = new int[2];

    string queryMIN = "SELECT MIN(No) FROM artist_net";
    string queryMAX = "SELECT MAX(No) FROM artist_net";

    MySqlCommand cmdMIN = new MySqlCommand(queryMIN, con);
    MySqlCommand cmdMAX = new MySqlCommand(queryMAX, con);

```

```

        try
        {
            con.Open();

            min = (int)cmdMIN.ExecuteScalar();
            max = (int)cmdMAX.ExecuteScalar();
            aggstore[0] = min;
            aggstore[1] = max;

            return aggstore;
        }
        finally
        {
            con.Close();
        }
    }

//Tar bort post(er) i DB:
public int DelRecord(int val)
{
    MySqlConnection con = new MySqlConnection(connectionString);
    int nodelrecs = 0;

    //Skapa kommandot
    string sql = "Delete from cd_net where Val=@Val";
    MySqlCommand cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@Val", val);

    try
    {
        con.Open();
        nodelrecs = cmd.ExecuteNonQuery();
        return nodelrecs;
    }
    finally
    {
        con.Close();
    }
}

//Uppdaterar post i DB:
public int UpdateRecord(int row, int artist_no, string album,
string format, string press,
                        string ar, string kommentar)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    //Skapa kommandot

```

```

        string sql = "update cd_net set Artist_no=@Artist_no,
Album=@Album, ";
        sql += "Format=@Format, Press=@Press, Ar=@Ar,
Kommentar=@Kommentar where `#`=@Row";

        MySqlCommand cmd = new MySqlCommand(sql, con);

        cmd.Parameters.AddWithValue("@Row", row);

        cmd.Parameters.AddWithValue("@Artist_no", artist_no);
        cmd.Parameters.AddWithValue("@Album", album);
        cmd.Parameters.AddWithValue("@Format", format);
        cmd.Parameters.AddWithValue("@Press", press);
        cmd.Parameters.AddWithValue("@Ar", ar);
        cmd.Parameters.AddWithValue("@Kommentar", kommentar);

        try
        {
            con.Open();
            int noupdrecs = cmd.ExecuteNonQuery();
            return noupdrecs;
        }
        finally
        {
            con.Close();
        }
    }

    //Söker i DB:
    public DataSet SearchRecords(string artist, string album, string
format, string press,
                                string ar, string kommentar)
    {
        string query = "select Artist, Album, Format, Press, Ar,
Kommentar, Val, `#`, Artist_no FROM cd_net, artist_net where Artist ";
        query += "in (select Artist from artist_net where Artist_no=No)
and Artist like @Artist and Album ";
        query += "like @Album and Format like @Format and Press like
@Press and Ar like @Ar ";
        query += "and Kommentar like @Kommentar order by album";

        MySqlCommand cmd = new MySqlCommand(query);

        cmd.Parameters.AddWithValue("@Artist", "%" + artist + "%");
        cmd.Parameters.AddWithValue("@Album", "%" + album + "%");
        cmd.Parameters.AddWithValue("@Format", "%" + format + "%");
        cmd.Parameters.AddWithValue("@Press", "%" + press + "%");
        cmd.Parameters.AddWithValue("@Ar", "%" + ar + "%");
        cmd.Parameters.AddWithValue("@Kommentar", "%" + kommentar +
"%");

        return FillDataSet(cmd, "Search");
    }

```

```

//Lämnar information om innehåll i skivtabell:
public OrderedDictionary GetInfo()
{
    MySqlConnection con = new MySqlConnection(connectionString);

    OrderedDictionary dictRec = new OrderedDictionary();

    //Ansamlar de aktuella formaten:
    string[] formatx = {"CD", "CDS", "CDM", "2CD", "LCDM", "XLCDM",
"DVDS",
    "CD+DVD", "DVD/SACD", "CDM/DVD", "dEP", "LP", "2LP", "BOX",
"12", "L12",
    "XL12", "L12num", "TwinSetL12", "TwinSetL12num", "Promo", "7",
"aEP",
    "mp3"};

    //Skapar Sql-satser:
    string[] sqlquery = new string[formatx.Length];

    for (int i = 0; i < formatx.Length; i++)
    {
        sqlquery[i] = "select count(Format) from cd_net where
Format='" + formatx[i] + "'";
    }

    //Skapa MySql-kommandon:
    MySqlCommand[] mysqlcmd = new MySqlCommand[formatx.Length];

    for (int i = 0; i < formatx.Length; i++)
    {
        mysqlcmd[i] = new MySqlCommand(sqlquery[i], con);
    }

    try
    {
        con.Open();

        //Exekverar kommandon samt adderar till kollektion:
        for (int i = 0; i < formatx.Length; i++)
        {
            dictRec.Add(formatx[i],
Convert.ToInt32((mysqlcmd[i]).ExecuteScalar()));
        }
        return dictRec;
    }
    finally
    { con.Close(); }
}

//Data till PieChart (på info-sidan):
public OrderedDictionary GetWholePie()

```

```

    {
        MySqlConnection con = new MySqlConnection(connectionString);

        //Dictionary<string, int> dictRec = new Dictionary<string,
int>();
        OrderedDictionary dictPieSlizes = new OrderedDictionary();

        //Ansamlar de aktuella musikstilarna:
        string[] musiktypx = {"Cyberpunk", "EBM", "EuroDisco",
"Gothrock", "HipHop",
"Industri", "New Wave", "Pop", "Punk", "RnB", "Rock", "Synth",
"Synthpop"};

        //Skapar Sql-satser:
        string[] sqlquery = new string[musiktypx.Length];

        for (int i = 0; i < musiktypx.Length; i++)
        {
            sqlquery[i] = "select count(Musiktyp) from artist_net where
Musiktyp='" + musiktypx[i] + "'";
        }

        //Skapa MySql-kommandon:
        MySqlCommand[] mysqlcmd = new MySqlCommand[musiktypx.Length];

        for (int i = 0; i < musiktypx.Length; i++)
        {
            mysqlcmd[i] = new MySqlCommand(sqlquery[i], con);
        }

        try
        {
            con.Open();

            //Exekverar kommandon samt adderar till kollektion:
            for (int i = 0; i < musiktypx.Length; i++)
            {
                dictPieSlizes.Add(musiktypx[i],
Convert.ToInt32((mysqlcmd[i]).ExecuteScalar()));
            }

            return dictPieSlizes;
        }
        finally
        { con.Close(); }
    }

    //Hämtar vald post för uppdatering:
    public DataSet FetchRecord()
    {

```



```

        string query = "SELECT Artist, Artist_no, Album, Format, Press,
Ar, Kommentar, Val, `#` FROM cd_net, ";
        query += "artist_net WHERE Artist IN (SELECT Artist FROM
artist_net WHERE Artist_no=No) and val=1;";
        MySqlCommand cmd = new MySqlCommand(query);

        return FillDataSet(cmd, "Fetch_row");
    }

```

//Användarkontrollsfunktioner:

```

//Visar en artist kort-biografi:
public DataSet CD_VisaKortBio(string row)
{
    string query = "select Artist, Kortbio from artist_net where
Artist=@Row";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@Row", row);

    return FillDataSet(cmd, "cd_visakortbio");
}

```

```

//Visar kommentar om en enhet:
public DataSet CD_Kommentarer(int row)
{
    string query = "select `#`,Kommentar FROM cd_net WHERE
`#`=@Row";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@Row", row);

    return FillDataSet(cmd, "cd_komtr");
}

```

```

//Visar fakta om en artist:
public DataSet CD_VisaArtistData(int row)
{
    string query = "select No, Artist, VPfr, Vptill, Ursland,
Ursstad, Musiktyp from artist_net where No=@Row";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@Row", row);
}

```

```

        return FillDataSet(cmd, "cd_visaartistdata");
    }

//Lägger till längre kommentar till DB:
public int AddKommentar(string kommentar, int kommentar_no)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    int noaddrescs = 0;

    //Skapa kommandot
    string sql = "update cd_net";
    sql += " set Kommentar=@Kommentar where `#`=@Kommentar_no";

    MySqlCommand cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@Kommentar", kommentar);
    cmd.Parameters.AddWithValue("@Kommentar_no", kommentar_no);

    try
    {
        con.Open();
        noaddrescs = cmd.ExecuteNonQuery();
        return noaddrescs;
    }
    finally
    {
        con.Close();
    }
}

//Hämtar längre kommentar från DB:
public DataSet FetchKommentar(int kommentar_no)
{
    string query = "SELECT Kommentar FROM cd_net WHERE
`#`=@kommentar_no";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@kommentar_no", kommentar_no);

    return FillDataSet(cmd, "Fetch_kommentar");
}

```

```
//Övriga funktioner:
```

```
//Exekverar alla funktioner (ovan) med en SELECT-sats:
private DataSet FillDataSet(MySqlCommand cmd, string tableName)
{
    MySqlConnection con = new MySqlConnection(connectionString);
    cmd.Connection = con;
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);
    DataSet ds = new DataSet();

    try
    {
        con.Open();
        adapter.Fill(ds, tableName);
    }
    finally
    {
        con.Close();
    }

    return ds;
}

}

}
```

Mellanskiktet - Komponent: DB_proc_Artist

Klass med olika funktioner gällande interaktion direkt mot databas för Artisttabellen
(börjar på nästa sida)

```

using System;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Configuration;
using System.Collections.Generic;
using System.Web.Caching;
using System.Collections;
using System.Collections.Specialized;

using MySql.Data.MySqlClient;

namespace DB_proc_Artist
{
    public class Artist
    {
        //Anslutning mot DB:

        //Anslutningssträng mot DB
        private string connectionString;

        //Anslutning mot DB inkl felhantering:
        public Artist()
        {
            if (WebConfigurationManager.ConnectionStrings["lokal"] == null)
            {
                throw new ApplicationException("ConnectionString saknas i
web.config.");
            }
            else
            {
                connectionString =
WebConfigurationManager.ConnectionStrings["lokal"].ConnectionString;
            }
        }

        //Allmänna DB-funktioner för "knapptsatsen":

        //Visa hela DB:
        public DataSet GetWholeDB()
        {
            string query = "select No, ValArt ,Artist, VPfr, VPtill,
Ursland, Ursstad, Musiktyp, Kortbio  from artist_net order by Artist asc";

            MySqlCommand cmd = new MySqlCommand(query);

            return FillDataSet(cmd, "whole");
        }
    }
}

```

```

        //Lägga till post i DB:
        public int AddRecord(int val, string artist, string text9_VPfr,
string text10_VPtill,
                                string text11_Ursland, string text12_Ursstad,
string text13_Musiktyp,
                                string text14_Kortbio)
        {
            MySqlConnection con = new MySqlConnection(connectionString);

            int noaddrecs = 0;

            //Skapa kommandot
            string sql = "insert into artist_net ";
            sql += "(ValArt, Artist, VPfr, VPtill, Ursland, Ursstad,
Musiktyp, Kortbio)";
            sql += " values(@ValArt, @Artist, @VPfr, @VPtill, @Ursland,
@Ursstad, @Musiktyp, @Kortbio)";
            MySqlCommand cmd = new MySqlCommand(sql, con);
            cmd.Parameters.AddWithValue("@ValArt", val);
            cmd.Parameters.AddWithValue("@Artist", artist);
            cmd.Parameters.AddWithValue("@VPfr", text9_VPfr);
            cmd.Parameters.AddWithValue("@VPtill", text10_VPtill);
            cmd.Parameters.AddWithValue("@Ursland", text11_Ursland);
            cmd.Parameters.AddWithValue("@Ursstad", text12_Ursstad);
            cmd.Parameters.AddWithValue("@Musiktyp", text13_Musiktyp);
            cmd.Parameters.AddWithValue("@Kortbio", text14_Kortbio);

            try
            {
                con.Open();
                noaddrecs = cmd.ExecuteNonQuery();
                return noaddrecs;
            }
            finally
            {
                con.Close();
            }
        }
    }

```

```

//Tar bort post(er) i DB:
public int DelRecord(int val)
{
    MySqlConnection con = new MySqlConnection(connectionString);
    int nodelrecs = 0;

    //Skapa kommandot
    string sql = "Delete from artist_net where ValArt=@ValArt";
    MySqlCommand cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@ValArt", val);

    try
    {
        con.Open();
        nodelrecs = cmd.ExecuteNonQuery();
        return nodelrecs;
    }
}

```

```

    }
    finally
    {

        con.Close();

    }
}

//Uppdaterar post i DB:
public int UpdateRecord(int row, string artist, string vpfr,
    string vptill, string ursland, string ursstad,
    string musiktyp, string kortbio)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    //Skapa kommandot
    string sql = "update artist_net set Artist=@Artist, Vpfr=@Vpfr,
";
    sql += "Vptill=@Vptill, Ursland=@Ursland, Ursstad=@Ursstad, ";
    sql += "Musiktyp=@Musiktyp, Kortbio=@Kortbio where No=@Row";

    MySqlCommand cmd = new MySqlCommand(sql, con);

    cmd.Parameters.AddWithValue("@Row", row);

    cmd.Parameters.AddWithValue("@Artist", artist);
    cmd.Parameters.AddWithValue("@Vpfr", vpfr);
    cmd.Parameters.AddWithValue("@Vptill", vptill);
    cmd.Parameters.AddWithValue("@Ursland", ursland);
    cmd.Parameters.AddWithValue("@Ursstad", ursstad);
    cmd.Parameters.AddWithValue("@Musiktyp", musiktyp);
    cmd.Parameters.AddWithValue("@Kortbio", kortbio);

    try
    {
        con.Open();
        int noupdrecs = cmd.ExecuteNonQuery();
        return noupdrecs;
    }
    finally
    {
        con.Close();
    }
}

```

```

//Söker i DB:

```

```

        public DataSet SearchRecords(string artist, string vpfr, string
vptill, string ursland,
                                string ursstad, string musiktyp, string kortbio)
        {
            string query = "select No, ValArt, Artist, VPfr, VPtill,
Ursland, Ursstad, Musiktyp FROM artist_net ";
            query += "where Artist like @Artist and VPfr ";
            query += "like @VPfr and VPtill like @VPtill and Ursland like
@Ursland and Ursstad like @Ursstad ";
            query += "and Musiktyp like @Musiktyp and Kortbio like @Kortbio
order by Artist";

            MySqlCommand cmd = new MySqlCommand(query);

            cmd.Parameters.AddWithValue("@Artist", "%" + artist + "%");
            cmd.Parameters.AddWithValue("@VPfr", "%" + vpfr + "%");
            cmd.Parameters.AddWithValue("@VPtill", "%" + vptill + "%");
            cmd.Parameters.AddWithValue("@Ursland", "%" + ursland + "%");
            cmd.Parameters.AddWithValue("@Ursstad", "%" + ursstad + "%");
            cmd.Parameters.AddWithValue("@Musiktyp", "%" + musiktyp + "%");
            cmd.Parameters.AddWithValue("@Kortbio", "%" + kortbio + "%");

            return FillDataSet(cmd, "Search");
        }

//Hämtar vald post för uppdatering:
public DataSet FetchRecord()
{
    string query="select * from artist_net where ValArt=1";

    MySqlCommand cmd = new MySqlCommand(query);

    return FillDataSet(cmd, "Fetch_row");
}

//Användarkontrollsfunktioner:

//Visar en artist kort-biografi:
public DataSet CD_VisaKortBio(string row)
{
    string query = "select Artist, Kortbio from artist_net where
Artist=@Row";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@Row", row);

    return FillDataSet(cmd, "cd_visakortbio");
}

```



```

//Visar fakta om en artist:
public DataSet CD_VisaArtistData(int row)
{
    string query = "select No, Artist, VPfr, Vptill, Ursland,
Ursstad, Musikttyp from artist_net where No=@Row";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@Row", row);

    return FillDataSet(cmd, "cd_visaartistdata");
}

//Adderar en kortbio-post till artisttabellen:
public int AddKortbio(string kortbio, int artist_no)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    int noaddrecs = 0;

    //Skapa kommandot
    string sql = "update artist_net";
    sql += " set Kortbio=@Kortbio where No=@No";

    MySqlCommand cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@Kortbio", kortbio);
    cmd.Parameters.AddWithValue("@No", artist_no);

    try
    {
        con.Open();
        noaddrecs = cmd.ExecuteNonQuery();
        return noaddrecs;
    }
    finally
    {
        con.Close();
    }
}

//Hämtar en kortbio från DB:
public DataSet FetchKortbio(int artist_no)
{
    string query = "SELECT Kortbio FROM artist_net WHERE No=@No";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@No", artist_no);

    return FillDataSet(cmd, "Fetch_kortbio");
}

```

```
//Övriga funktioner:
```

```
//Exekvererar alla funktioner (ovan) med en SELECT-sats:  
private DataSet FillDataSet(MySqlCommand cmd, string tableName)  
{  
    MySqlConnection con = new MySqlConnection(connectionString);  
    cmd.Connection = con;  
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);  
    DataSet ds = new DataSet();  
  
    try  
    {  
        con.Open();  
        adapter.Fill(ds, tableName);  
    }  
    finally  
    {  
        con.Close();  
    }  
  
    return ds;  
}  
  
}
```

Mellanskiktet - Komponent: DB_proc_Inkop

Klass med olika funktioner gällande interaktion direkt mot databas för Inkopstabellen
(börjar på nästa sida)

```

using System;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Configuration;
using System.Collections.Generic;
using System.Web.Caching;
using System.Collections;
using System.Collections.Specialized;

using MySql.Data.MySqlClient;

namespace DB_proc_Inkop
{
    public class Inkop
    {
        //Anslutning mot DB:

        //Anslutningssträng mot DB
        private string connectionString;

        //Anslutning mot DB inkl felhantering:
        public Inkop()
        {
            if (WebConfigurationManager.ConnectionStrings["lokal"] == null)
            {
                throw new ApplicationException("ConnectionString saknas i
web.config.");
            }
            else
            {
                connectionString =
WebConfigurationManager.ConnectionStrings["lokal"].ConnectionString;
            }
        }

        //Allmänna DB-funktioner för "knapptsatsen":

        //Visa hela DB:
        public DataSet GetWholeDB()
        {
            string query = "SELECT Skiv_no, ValInk , Art_no, Titel, Form,
Land, Utg, Komt, Inm_dat, Kop_grad, Kop_kat, CAST(Ca_pris AS UNSIGNED INT)
AS Ca_pris, No, Artist FROM inkop_net, artist_net WHERE No=Art_no ORDER BY
Artist, Titel asc";
            MySqlCommand cmd = new MySqlCommand(query);

```

```

        return FillDataSet(cmd, "whole");
    }

    //Lägga till post i DB:
    public int AddRecord(int val, int artist_no, string album, string
format,
    string press, string ar, string kommentar, string inm_dat, string
kop_grad,
    string kop_kat, string ca_pris)
    {
        MySqlConnection con = new MySqlConnection(connectionString);

        int noaddrescs = 0;

        //Skapa kommandot
        string sql = "insert into inkop_net";
        sql += "(ValInk, Art_no, Titel, Form, Land, Utg, Komt, Inm_dat,
";
        sql += "Kop_grad, Kop_kat, Ca_pris)";
        sql += " values(@ValInk, @Art_no, @Titel, @Form, @Land, @Utg,
@Komt, ";
        sql += "@Inm_dat, @Kop_grad, @Kop_kat, @Ca_pris)";
        MySqlCommand cmd = new MySqlCommand(sql, con);
        cmd.Parameters.AddWithValue("@ValInk", val);
        cmd.Parameters.AddWithValue("@Art_no", artist_no);
        cmd.Parameters.AddWithValue("@Titel", album);
        cmd.Parameters.AddWithValue("@Form", format);
        cmd.Parameters.AddWithValue("@Land", press);
        cmd.Parameters.AddWithValue("@Utg", ar);
        cmd.Parameters.AddWithValue("@Komt", kommentar);
        cmd.Parameters.AddWithValue("@Inm_dat", inm_dat);
        cmd.Parameters.AddWithValue("@Kop_grad", kop_grad);
        cmd.Parameters.AddWithValue("@Kop_kat", kop_kat);
        cmd.Parameters.AddWithValue("@Ca_pris", ca_pris);

        try
        {
            con.Open();
            noaddrescs = cmd.ExecuteNonQuery();
            return noaddrescs;
        }
        finally
        {
            con.Close();
        }
    }

    //Hämtar min- och maxgränser för rangevalidator gällande
begränsingar
    //av artisttabellens omfång (antal f.n inmatade artister)
    //Arbetar i samband med AddRecord-funktionen ovan:

```

```

public int[] GetAggCDNet()
{
    MySqlConnection con = new MySqlConnection(connectionString);
    int min = 0;
    int max = 0;
    int[] aggstore = new int[2];

    string queryMIN = "SELECT MIN(No) FROM artist_net";
    string queryMAX = "SELECT MAX(No) FROM artist_net";

    MySqlCommand cmdMIN = new MySqlCommand(queryMIN, con);
    MySqlCommand cmdMAX = new MySqlCommand(queryMAX, con);

    try
    {
        con.Open();

        min = (int)cmdMIN.ExecuteScalar();
        max = (int)cmdMAX.ExecuteScalar();
        aggstore[0] = min;
        aggstore[1] = max;

        return aggstore;
    }
    finally
    {
        con.Close();
    }
}

//Tar bort post(er) i DB:
public int DelRecord(int val)
{
    MySqlConnection con = new MySqlConnection(connectionString);
    int nodelrecs = 0;

    //Skapa kommandot
    string sql = "Delete from inkop_net where ValInk=@ValInk";
    MySqlCommand cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@ValInk", val);

    try
    {
        con.Open();
        nodelrecs = cmd.ExecuteNonQuery();
        return nodelrecs;
    }
    finally
    {
        con.Close();
    }
}

```

```

    }

    //Uppdaterar post i DB:
    public int UpdateRecord(int row, int artist_no, string album,
string format, string press,
                                string ar, string kommentar, string
inm_dat, string kop_grad,
                                string Kop_kat, string Ca_pris)
    {
        MySqlConnection con = new MySqlConnection(connectionString);

        //Skapa kommandot
        string sql = "update inkop_net set Art_no =@Art_no,
Titel=@Titel, ";
        sql += "Form=@Form, Land=@Land, Utg=@Utg, Komt=@Komt, ";
        sql += "Inm_dat=@Inm_dat, Kop_grad=@Kop_grad, Kop_kat=@Kop_kat,
";

        sql += "Ca_pris=@Ca_pris where Skiv_no=@Row";

        MySqlCommand cmd = new MySqlCommand(sql, con);

        cmd.Parameters.AddWithValue("@Row", row);

        cmd.Parameters.AddWithValue("@Art_no", artist_no);
        cmd.Parameters.AddWithValue("@Titel", album);
        cmd.Parameters.AddWithValue("@Form", format);
        cmd.Parameters.AddWithValue("@Land", press);
        cmd.Parameters.AddWithValue("@Utg", ar);
        cmd.Parameters.AddWithValue("@Komt", kommentar);
        cmd.Parameters.AddWithValue("@Inm_dat", inm_dat);
        cmd.Parameters.AddWithValue("@Kop_grad", kop_grad);
        cmd.Parameters.AddWithValue("@Kop_kat", Kop_kat);
        cmd.Parameters.AddWithValue("@Ca_pris", Ca_pris);

        try
        {
            con.Open();
            int noupdrecs = cmd.ExecuteNonQuery();
            return noupdrecs;
        }
        finally
        {
            con.Close();
        }
    }

    //Söker i DB:
    public DataSet SearchRecords(string art_no, string album, string
format, string press,
                                string ar, string kommentar, string inm_dat, string
kop_grad,
                                string kop_kat, string ca_pris)
    {

```

```

        string query = "select Skiv_no, ValInk, Art_no, Titel,
Form, Land, Utg, Komt, Inm_dat, Kop_grad, ";
        query += "Kop_kat, CAST(Ca_pris AS UNSIGNED INT) AS
Ca_pris, Artist FROM inkop_net, artist_net where Artist ";
        query += "in (select Artist from artist_net where
Art_no=No) and Artist like @Artist and Titel ";
        query += "like @Titel and Form like @Form and Land like
@Land and Utg like @Utg ";
        query += "and Komt like @Komt and Inm_dat like @Inm_dat and
Kop_grad like @Kop_grad and ";
        query += "Kop_kat like @Kop_kat and Ca_pris like @Ca_pris
order by Titel";

```

```

        MySqlCommand cmd = new MySqlCommand(query);

```

```

        cmd.Parameters.AddWithValue("@Artist", "%" + art_no + "%");
        cmd.Parameters.AddWithValue("@Titel", "%" + album + "%");
        cmd.Parameters.AddWithValue("@Form", "%" + format + "%");
        cmd.Parameters.AddWithValue("@Land", "%" + press + "%");
        cmd.Parameters.AddWithValue("@Utg", "%" + ar + "%");
        cmd.Parameters.AddWithValue("@Komt", "%" + kommentar + "%");
        cmd.Parameters.AddWithValue("@Inm_dat", "%" + inm_dat + "%");
        cmd.Parameters.AddWithValue("@Kop_grad", "%" + kop_grad + "%");
        cmd.Parameters.AddWithValue("@Kop_kat", "%" + kop_kat + "%");
        cmd.Parameters.AddWithValue("@Ca_pris", "%" + ca_pris + "%");

```

```

        return FillDataSet(cmd, "Search");

```

```

    }

```

```

//Hämtar vald post för uppdatering:

```

```

public DataSet FetchRecord()

```

```

{

```

```

        string query = "SELECT Skiv_no, ValInk, Art_no, Titel, Form,
Land, Utg, Komt, Inm_dat, Kop_grad, Kop_kat, ";
        query += "Ca_pris, Artist FROM inkop_net, ";
        query += "artist_net WHERE Artist IN (SELECT Artist FROM
artist_net WHERE Art_no=No) and ValInk=1;";
        MySqlCommand cmd = new MySqlCommand(query);

```

```

        return FillDataSet(cmd, "Fetch_row");

```

```

    }

```

```

//Överför post från Inkopstabell till Skivtabell:

```

```

public int TransferRecord(int valink)

```

```

{

```

```

        MySqlConnection con = new MySqlConnection(connectionString);
        int notransfrecs = 0;

```



```

        //Väljer ut post i inkoptabell och kopierar den till
skivtabell:
        string sql = "insert into cd_net";
        sql += "(Artist_no, Album, Format, Press, Ar, Kommentar) ";
        sql += "select Art_no, Titel, Form, Land, Utg, Komt ";
        sql += "from inkop_net where ValInk=@ValInk";

        //Tar bort överförd post i inkopstabell:
        string sql2 = "delete from inkop_net where ValInk=1";

        MySqlCommand cmd = new MySqlCommand(sql, con);
        MySqlCommand cmd2 = new MySqlCommand(sql2, con);

        cmd.Parameters.AddWithValue("@ValInk", valink);

        try
        {
            con.Open();
            notransfrecs = cmd.ExecuteNonQuery();
            cmd2.ExecuteNonQuery();

            return notransfrecs;
        }
        finally
        {
            con.Close();
        }
    }
}

```

//Användarkontrollsfunktioner:

```

//Visar en artist kort-biografi:
public DataSet CD_VisaKortBio(string row)
{
    string query = "select Artist, Kortbio from artist_net where
Artist=@Row";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@Row", row);

    return FillDataSet(cmd, "cd_visakortbio");
}

//Visar kommentar om en enhet:
public DataSet CD_Kommentarer(int row)

```

```

    {
        string query = "select Skiv_no, Komt FROM inkop_net WHERE
Skiv_no =@Row";

        MySqlCommand cmd = new MySqlCommand(query);
        cmd.Parameters.AddWithValue("@Row", row);

        return FillDataSet(cmd, "inkop_komtr");
    }

//Visar fakta om en artist:
public DataSet CD_VisaArtistData(int row)
{
    string query = "select No, Artist, VPfr, Vptill, Ursland,
Ursstad, Musiktyp from artist_net where No=@Row";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@Row", row);

    return FillDataSet(cmd, "inkop_visaartistdata");
}

//Lägger till längre kommentar till DB:
public int AddKommentar(string kommentar, int kommentar_no)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    int noaddrecs = 0;

    //Skapa kommandot
    string sql = "update inkop_net";
    sql += " set Komt=@Komt where Skiv_no=@Kommentar_no";

    MySqlCommand cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@Komt", kommentar);
    cmd.Parameters.AddWithValue("@Kommentar_no", kommentar_no);

    try
    {
        con.Open();
        noaddrecs = cmd.ExecuteNonQuery();
        return noaddrecs;
    }
    finally
    {
        con.Close();
    }
}

```

```

//Hämtar längre kommentar från DB:
public DataSet FetchKommentar(int kommentar_no)
{
    string query = "SELECT Komt FROM inkop_net WHERE
Skiv_no=@kommentar_no";

    MySqlCommand cmd = new MySqlCommand(query);
    cmd.Parameters.AddWithValue("@kommentar_no", kommentar_no);

    return FillDataSet(cmd, "Fetch_kommentar");
}

//Övriga funktioner:

//Exekverar alla funktioner (ovan) med en SELECT-sats:
private DataSet FillDataSet(MySqlCommand cmd, string tableName)
{
    MySqlConnection con = new MySqlConnection(connectionString);
    cmd.Connection = con;
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);
    DataSet ds = new DataSet();

    try
    {
        con.Open();
        adapter.Fill(ds, tableName);
    }
    finally
    {
        con.Close();
    }

    return ds;
}

}

```

Mellanskiktet - Komponent: Common_Tasks

Innehåller två klasser med funktioner för dels göra icke-aktuella element för aktuell process osynliga (invisible) exempelvis för en utskrift samt en klass som sköter hanteringen av kryssrutor (i gridviews o.dyl) direkt mot databas. Detta gällande för Skivtabellen.

Gör icke-aktuella element för aktuell process osynliga (invisible) exempelvis för en utskrift.
Detta gällande för Skivtabellen.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI;

namespace Common_Tasks
{
    public class Clear
    {
        public void Clean_surfaces(Page sida)
        {
            UserControl grd_Artistdata_UCtrl =
            (UserControl)sida.FindControl("grd_Artistdata_UCtrl");
            UserControl grdResult_UCtrl =
            (UserControl)sida.FindControl("grdResult_UCtrl");
            UserControl gridArtister_UCtrl =
            (UserControl)sida.FindControl("gridArtister_UCtrl");
            UserControl pnlKommentar_UCtrl =
            (UserControl)sida.FindControl("pnlKommentar_UCtrl");

            GridView grd_Artistdata =
            (GridView)grd_Artistdata_UCtrl.FindControl("grd_Artistdata");
            GridView grdResult =
            (GridView)grdResult_UCtrl.FindControl("grdResult");
            GridView gridArtister =
            (GridView)gridArtister_UCtrl.FindControl("gridArtister");
            Panel pnlKommentar =
            (Panel)pnlKommentar_UCtrl.FindControl("pnlKommentar");

            //Chart ch3DPie = (Chart)sida.FindControl("ch3DPie");
            //Label lbl3DPie = (Label)sida.FindControl("lbl3DPie");

            HtmlGenericControl display =
            (HtmlGenericControl)sida.FindControl("display");
            HtmlGenericControl display2 =
            (HtmlGenericControl)sida.FindControl("display2");

            gridArtister.Visible = false;
            grdResult.Visible = false;
            grd_Artistdata.Visible = false;
            pnlKommentar.Visible = false;
            //ch3DPie.Visible = false;
            //lbl3DPie.Visible = false;

            display.InnerHtml = "";
            display2.InnerHtml = "";
        }
    }
}
```

} }

Sköter hanteringen av kryssrutor (i gridviews o.dyl) direkt mot databas. Detta gällande för Skivtabellen.

```
using System;
using System.Collections.Generic;
using System.Text;
using MySql.Data.MySqlClient;
using System.Web.Configuration;
using System.Web.UI.WebControls;

using DB_proc;
using TF.Namespace.Controls;

namespace Common_Tasks
{
    public class SetChks
    {
        private string connectionString;

        //Konstruktorn möjliggör för säker anslutning mot DB via
        web.config-inställningarna:
        public SetChks()
        {
            if (WebConfigurationManager.ConnectionStrings["lokal"] == null)
            {
                throw new ApplicationException("ConnectionString saknas i
web.config.");
            }
            else
            {
                connectionString =
WebConfigurationManager.ConnectionStrings["lokal"].ConnectionString;
            }
        }

        //På- och avmarkera enskilda kryssrutor i skivtabell samt
        skivsökresultat-överlagrad:
        public void UpdateRecord_ChkBox(int row, int val)
        {
            MySqlConnection con = new MySqlConnection(connectionString);

            //Skapa kommandot
            string sql = "update cd_net set Val=@Val where `#`=@Row";
            MySqlCommand cmd = new MySqlCommand(sql, con);
            cmd.Parameters.AddWithValue("@Val", val);
            cmd.Parameters.AddWithValue("@Row", row);

            try
            {
                con.Open();
                cmd.ExecuteNonQuery();
            }
            finally
            {
```

```

        {
            con.Close();
        }
    }

    //På- och avmarkera alla kryssrutor i skivtabell, även vid
    skivrefresh-överlagrad:
    public void UpdateRecord_ChkBox(int val)
    {
        MySqlConnection con = new MySqlConnection(connectionString);

        //Skapa kommandot
        string sql = "update cd_net set Val=@Val";
        MySqlCommand cmd = new MySqlCommand(sql, con);
        cmd.Parameters.AddWithValue("@Val", val);

        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
        }
        finally
        {
            con.Close();
        }
    }
}

```


Mellanskiktet - Komponent: Common_Tasks_Artist

Innehåller två klasser med funktioner för dels göra icke-aktuella element för aktuell process osynliga (invisible) exempelvis för en utskrift samt en klass som sköter hanteringen av kryssrutor (i gridviews o.dyl) direkt mot databas. Detta gällande för Artisttabellen.

Gör icke-aktuella element för aktuell process osynliga (invisible) exempelvis för en utskrift. Detta gällande för Artisttabellen.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI;

namespace Common_Tasks_Artist
{
    public class Clear_Artist
    {
        public void Clean_surfaces_Artist(Page sida)
        {
            UserControl gridArtister_Artist_UCtrl =
            (UserControl)sida.FindControl("gridArtister_Artist_UCtrl");
            UserControl grdResult_Artist_UCtrl =
            (UserControl)sida.FindControl("grdResult_Artist_UCtrl");
            UserControl pnlKortbio_UCtrl =
            (UserControl)sida.FindControl("pnlKortbio_UCtrl");

            GridView gridArtister_Artist =
            (GridView)gridArtister_Artist_UCtrl.FindControl("gridArtister_Artist");
            GridView grdResult_Artist =
            (GridView)grdResult_Artist_UCtrl.FindControl("grdResult_Artist");
            Panel pnlKortbio =
            (Panel)pnlKortbio_UCtrl.FindControl("pnlKortbio");

            HtmlGenericControl display =
            (HtmlGenericControl)sida.FindControl("display");
            HtmlGenericControl display2 =
            (HtmlGenericControl)sida.FindControl("display2");

            gridArtister_Artist.Visible = false;
            grdResult_Artist.Visible = false;
            pnlKortbio.Visible = false;

            display.InnerHtml = "";
            display2.InnerHtml = "";
        }
    }
}
```

Sköter hanteringen av kryssrutor (i gridviews o.dyl) direkt mot databas. Detta gällande för Artisttabellen.

```
using System;
using System.Collections.Generic;
using System.Text;
using MySql.Data.MySqlClient;
using System.Web.Configuration;
using System.Web.UI.WebControls;

using DB_proc_Artist;
using TF.Namespace.Controls;

namespace Common_Tasks_Artist
{
    public class SetChks_Artist
    {
        private string connectionString;

        //Konstruktorn möjliggör för säker anslutning mot DB via
        web.config-inställningarna:
        public SetChks_Artist()
        {
            if (WebConfigurationManager.ConnectionStrings["lokal"] == null)
            {
                throw new ApplicationException("ConnectionString saknas i
web.config.");
            }
            else
            {
                connectionString =
WebConfigurationManager.ConnectionStrings["lokal"].ConnectionString;
            }
        }

        //På- och avmarkera enskilda kryssrutor i artisttabell samt
        artistsökresultat-överlagrad:
        public void UpdateRecord_Artist_ChkBox(int row, int val)
        {
            MySqlConnection con = new MySqlConnection(connectionString);

            //Skapa kommandot
            string sql = "update artist_net set ValArt=@Val where No=@Row";
            MySqlCommand cmd = new MySqlCommand(sql, con);
            cmd.Parameters.AddWithValue("@Val", val);
            cmd.Parameters.AddWithValue("@Row", row);

            try
            {
                con.Open();
                cmd.ExecuteNonQuery();
            }
            finally
            {
                con.Close();
            }
        }
    }
}
```

```

    }
}

//På- och avmarkera alla kryssrutor i artisttabell, även vid
artistrefresh-överlagrad:
public void UpdateRecord_Artist_ChkBox(int val)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    //Skapa kommandot
    string sql = "update artist_net set ValArt=@Val";
    MySqlCommand cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@Val", val);

    try
    {
        con.Open();
        cmd.ExecuteNonQuery();
    }
    finally
    {
        con.Close();
    }
}

}
}

```

Mellanskiktet - Komponent: Common_Tasks_Inkop

Innehåller två klasser med funktioner för dels göra icke-aktuella element för aktuell process osynliga (invisible) exempelvis för en utskrift samt en klass som sköter hanteringen av kryssrutor (i gridviews o.dyl) direkt mot databas. Detta gällande för Inkopstabellen.

Gör icke-aktuella element för aktuell process osynliga (invisible) exempelvis för en utskrift. Detta gällande för Inkopstabellen.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI;

namespace Common_Tasks_Inkop
{
    public class Clear_Inkop
    {
        public void Clean_surfaces_Inkop(Page sida)
        {
            UserControl UGridArtister_Inkop_UCtrl =
            (UserControl)sida.FindControl("UGridArtister_Inkop_UCtrl");
            UserControl grdResult_Inkop_UCtrl =
            (UserControl)sida.FindControl("grdResult_Inkop_UCtrl");
            UserControl grd_Artistdata_Inkop_UCtrl =
            (UserControl)sida.FindControl("grd_Artistdata_Inkop_UCtrl");
            UserControl pnlKommentar_Inkop_UCtrl =
            (UserControl)sida.FindControl("pnlKommentar_Inkop_UCtrl");

            GridView gridArtister_Inkop =
            (GridView)UGridArtister_Inkop_UCtrl.FindControl("gridArtister_Inkop");
            GridView grdResult_Inkop =
            (GridView)grdResult_Inkop_UCtrl.FindControl("grdResult_Inkop");
            GridView grd_Artistdata_Inkop =
            (GridView)grd_Artistdata_Inkop_UCtrl.FindControl("grd_Artistdata_Inkop");
            Panel pnlKommentar_Inkop =
            (Panel)pnlKommentar_Inkop_UCtrl.FindControl("pnlKommentar_Inkop");

            HtmlGenericControl display =
            (HtmlGenericControl)sida.FindControl("display");
            HtmlGenericControl display2 =
            (HtmlGenericControl)sida.FindControl("display2");

            gridArtister_Inkop.Visible = false;
            grdResult_Inkop.Visible = false;
            grd_Artistdata_Inkop.Visible = false;
            pnlKommentar_Inkop.Visible = false;

            display.InnerHtml = "";
            display2.InnerHtml = "";
        }
    }
}
```

Sköter hanteringen av kryssrutor (i gridviews o.dyl) direkt mot databas. Detta gällande för Inkopstabellen.

```
using System;
using System.Collections.Generic;
using System.Text;
using MySql.Data.MySqlClient;
using System.Web.Configuration;
using System.Web.UI.WebControls;

using DB_proc_Inkop;
using TF.Namespace.Controls;

namespace Common_Tasks_Inkop
{
    public class SetChks_Inkop
    {
        private string connectionString;

        //Konstruktorn möjliggör för säker anslutning mot DB via
        web.config-inställningarna:
        public SetChks_Inkop()
        {
            if (WebConfigurationManager.ConnectionStrings["lokal"] == null)
            {
                throw new ApplicationException("ConnectionString saknas i
web.config.");
            }
            else
            {
                connectionString =
WebConfigurationManager.ConnectionStrings["lokal"].ConnectionString;
            }
        }

        //På- och avmarkera enskilda kryssrutor i inkoptabell samt
        inkopsökresultat-överlagrad:
        public void UpdateRecord_ChkBox(int row, int val)
        {
            MySqlConnection con = new MySqlConnection(connectionString);

            //Skapa kommandot
            string sql = "update inkop_net set ValInk =@ValInk where
Skiv_no =@Row";
            MySqlCommand cmd = new MySqlCommand(sql, con);
            cmd.Parameters.AddWithValue("@ValInk", val);
            cmd.Parameters.AddWithValue("@Row", row);

            try
            {
                con.Open();
                cmd.ExecuteNonQuery();
            }
            finally
            {
                con.Close();
            }
        }
    }
}
```

```

        con.Close();
    }
}

//På- och avmarkera alla kryssrutor i inkoptabell, även vid
inkoprefresh-överlagrad:
public void UpdateRecord_ChkBox(int val)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    //Skapa kommandot
    string sql = "update inkop_net set ValInk=@ValInk";
    MySqlCommand cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@ValInk", val);

    try
    {
        con.Open();
        cmd.ExecuteNonQuery();
    }
    finally
    {
        con.Close();
    }
}

}

}

```


Mellanskiktet - Komponent: CtrlLib

Komponent som innehåller klasser där befintliga kontroller moduleras. I skrivande stund är det endast en kontroll, en kryssruta, som förlängs/moduleras med ytterligare en egenskap/attribut (property) som skall underlätta lagring av värden.

```

using System;
using System.ComponentModel;
using System.Security;
using System.Security.Permissions;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

//Custom checkbox som har uttökats med ett extra "lagringsattribut":
UnText. Som Text fast ej
//synlig i browseren.

namespace TF.Namespace.Controls
{
    public class DerivCheckBox : CheckBox
    {
        public string UnText
        {
            get { string un = (string)ViewState["UnText"]; return un; }
            set{ ViewState["UnText"] = value; }
        }
    }
}

```

BACKEND

Backend - MySQL-databas

Beskrivningen av denna databas samt några ord om hur den fungerar i just min applikation skall jag inte göra här då det har skett under ett antal andra rubriker i denna dokumentation. Dessa är:

- Installation-databas
- Databasstruktur
- Normaliseringsformer
- Optimering
- och även under rubriken "I användning"

....jag hänvisar därför Er till dessa rubriker istället.

Men kan väl ändå här och nu nämna att databasen ifråga i skrivande stund är en MySQL-databas med versionsbeteckningen ” 5.2.5.0”. Detta kan självklart ändras snabbt främst beroende på vad som stöds och används av mitt webbhotell där applikationen för närvarande huserar.

Bilagor

Bilaga 1 - Testschema

Nedanstående testschema är det som jag använt vid mina tester av denna applikation. Dess främsta syfte är att jag har kunnat på ett mer metodiskt sätt göra genomgående tester av applikationens funktioner. Inte minst har den gjort det lättare att inte missa någon funktionalitet. Testschemat är sammanställt parallellt med ordinarie programmeringsarbete och därför mest avsett för mig själv och därmed inte särskilt användarvänligt strukturerat. Men jag tog med det ändå för att lättare kunna ge en helhetsbild av det arbete som har fortgått samt även påvisa den grad av seriositet som jag ger mina projekt. Inget seriöst webbutvecklings-projekt värt namnet är ju utan ordentliga tester, eller hur?

Testschema för asp.net+c# webbdatabas:

- Visa hel databas
- Testa paging
(Både för sökresultat och hel tabell)
- Testa ikryssning av headerkryssruta samt enskilda kryssrutor + kolla i databas så ändring verkligen sker.
(Både för sökresultat och hel tabell)
- Testa så att ikryssade enskilda kryssrutor förblir ikryssade vid paging.
(Både för sökresultat och hel tabell)
- Lägg till en post
- Ta bort en post
(Både för sökresultat och hel tabell)
- Uppdatera en post
(Både för sökresultat och hel tabell)
- Hämta en post för uppdatering (fetch)
(Både för sökresultat och hel tabell)
- Kolla så rensning av ikryssade kryssrutor efter uppdatering fungerar
(Både för sökresultat och hel tabell)
- Kolla så info om aktuell tabell fungerar ordentligt med rätt info, antal, sort etc (Skivtabell)
- kolla så Omstart_TB återstartar applikationen
- kolla så Börja_om rensar alla rutor
- kolla exception för lägg till (överstigning av högsta värde, ej matat in artistnummer, endast siffror är tillåtna). OBS olika validering och därmed exceptions för olika tabeller
- kolla exception för ta bort (minst en kryssruta måste vara ifylld)
- kolla exception för hämta (ingen kryssruta ifylld, endast EN kryssruta får vara ifylld)
(Både för sökresultat och hel tabell)
- Testa sökning + kolla så cachan fungerar

- Testa kortbiopanel (Visa själva panelen, hämta, mata in, rensa)
 - Testa Kommentarpanel (Visa själva panelen, hämta, mata in, rensa)
 - Testa visa artistdata
(Både för sökresultat och hel tabell)
 - Testa visa en artists kortbio
(Både för sökresultat och hel tabell)
 - Testa kommentarknapp (där sådan finnes)
(Både för sökresultat och hel tabell)
 - Testa Överför (Inkopstabel)
- Testa exceptions för Överför (Ej kryssat i någon ruta, kryssat i mer än en ruta)
(Både för sökresultat och hel tabell)
- Testa valideringen för Inm_data-textrutan (Inkopstabel)
 - Testa valideringen för Cpris-textrutan (Inkopstabel)
 - Testa validering för Ursstad-textrutan (Artisttabell)
 - Testa de olika dynamiskt genererade sorteringsfunktionerna för de fält i aktuell gridview som har dem, så att de fungerar som de skall
(Både för sökresultat och hel tabell)

+ Samt olika tester av olika funktioners catch-satser (FormatException, och System.Exception)

Genererar fel för test:

```
throw new FormatException();
throw new System.Exception();
```

(MySqlException FUNKAR EJ I VER 5.2.5 (ej några def konstruktörer))

Testschema för olika funktioners FormatExceptions och System.Exceptions:

Callobjekt	DBAccessKomp-metod (cs-fil)	Browser Chrome	Browser IE	Browser FF
Page:				
Lagg_till	AddRecord (Proc_act.cs)			
Ta_bort	DelRecord (Proc_act.cs)			

Uppdatera	UpdateRecord (Proc_act.cs)			
Sök	SearchRecords (Proc_act.cs)			
Info	GetInfo (Proc_act.cs)			
Kortbio (Visa Kortbiopanel)	None (Visa_KortBioPanel.cs)			
Kommentar (Visa Kommentarpanel)	None (Visa_Kommentarpanel.cs)			
Fetch (Hämta)	FetchRecord (Proc_act.cs)			
Refresh_TB (Omstart_TB)	UpdateRecord_ChkBox ALLA kryssrutor. Överlagrad (SetChks.cs)			
UCtrl:				
grd_Artistdata	CD_VisaKortBio (Proc_act.cs)			
GrdView_grdResult	CD_Kommentar (Proc_act.cs)			
	CD_VisaArtistData (Proc_act.cs)			
	UpdateRecord_ChkBox Enskilda kryssrutor. Överlagrad (SetChks.cs)			
	UpdateRecord_ChkBox ALLA kryssrutor (head). Överlagrad (SetChks.cs)			
Gridview_gridArtister	CD_Kommentar (Proc_act.cs)			
	CD_VisaArtistData (Proc_act.cs)			
	UpdateRecord_ChkBox Alla kryssrutor (head). Överlagrad (SetChks.cs)			
	UpdateRecord_ChkBox Enskilda kryssrutor. Överlagrad (SetChks.cs)			
	GetWholeDB (Proc_act.cs)			
Panel_Kortbio	AddKortbio (Proc_act.cs)			

	FetchKortbio (Proc_act.cs)			
Panel_Kommentar	AddKommentar (Proc_act.cs)			
	FetchKommentar (Proc_act.cs)			

Bilaga 2 – Format på olika skivor/media som kan matas in i databasen

Naturligtvis kan denna databas med vissa få modifieringar mata in data om media som lagrar vilket ljudformat som helst. Men de som jag har angett från början är ett flertal cd/dvd- och vinylformat samt ljuddataformatet mp3 som är ett mycket populärt format när det gäller att köpa musik på Internet. Jag nöjde mig med detta ljuddataformat då det är det för närvarande dominerande men kan som sagt var lägga till andra eller fler. Med uppräkningsen av dessa format följer en kort beskrivning då vissa av formaten och/eller beteckningarna kan te sig aningen suspekta och en del är egensammansatta kombinationer som även de kan bytas ut mot något annat önskvärt.

Det kan hända att fler lagringsformat har lagts till i applikationen efter det här har skrivits än vad som nämns nedan. Men jag tror ändå att det skall inkludera flertalet av formaten och ge en känsla för vad som finns. För format som kan förekomma men inte finns här och om intresse finns att ta reda på mer om dessa, så slå upp dem på Internet där finns mycket information om olika format för lagring av ljud (och även bild).

CD

Compact Disc, det vanligaste formatet idag för att lagra digital musik även om det har hängt med längre och det ständigt kommer konkurrenter. Ett fullständigt album digitalt lagrat med andra ord.

CDM

Compact Disc Maxisingle. som CD fast i singelutförande men innehåller fler låtar än singelns vanligtvis endast två.

CDS

Compact Disc Single, som CDM fast ofta med endast två spår.

2CD

Double Compact Disc, som CD fast med två CD skivor.

LCDM

Limited Compact Disc Maxisingle, som CDM fast utgiven i en begränsad upplaga.

XLCDM

Extra Limited Compact Disc Maxisingle, som LCDM fast utgiven i en än mer begränsad upplaga.

DVDS

Digital Versatile Disc Single. DVD skivor kommer med sin högre upplösning även starkt inom musiken och inte bara för filmbruk. Sådana här singlar brukar oftast förutom ljudmaterial även innehålla musikvideon som representerar singellåten.

CD+DVD

En kombination där vanligtvis CD-skivan innehåller ljudmaterial i vanlig ordning medans DVD-skivan innehåller bildmaterial samt samma ljudmaterial som CD:n fast kanske i exempelvis en surroundmix. En vanligt konfiguration är en livekonsert där konserten är filmad och ligger på DVD och där konsertens låtar även ligger på CD-skivan.

DVD/SACD

Liknande CD+DVD med skillnaden att CD-skivan är utgiven i SACD (Super Audio Compact Disc) formatet som man enkelt kan uttrycka är förbättringen av CD-formatet.

CDM/DVD

En kombination som liknar CD+DVD fast istället för ett fullt album är det en cd-single.

dEP

Digital extended play. EP är som ett album fast mindre antal låtar, en till omfånget mindre CD kan man säga. Bokstaven d som står innan är en egen benämning för att här i databasen särskilja mellan analog och digital EP.

LP

Long Play, föregångaren till CD där inspelningen har skett analogt. Ett fullständigt analogt lagrat album med andra ord.

2LP

Double Long Play, som LP fast innehållandes två LP-skivor.

BOX

En box som innehåller oftast ett antal LP-skivor och inte sällan heller andra bonussaker som filmmaterial, bildhäfte med mera.

12

Tolvan står för antalet tum som denna singels diameter och är vad som i folkmun kallas för en maxisingel. Det vill säga en singel med fler antal låtar än en "vanlig" singel, se även CDM ovan.

L12

Limited 12, som 12 fast utgiven i en begränsad upplaga.

XL12

Extra Limited 12, som L12 fast utgiven i en än mer begränsad upplaga.

L12num

Limited 12 numbered, som L12 fast där den begränsade upplagen är numrerad, och där lägre nummer dvs tidigare utgivna är hett eftertraktade av samlare.

TwinSetL12

TwinSet Limited 12, en begränsad utgåva där två stycken 12 tums maxisinglar ingår.

TwinSetL12num

TwinSet Limited 12 numbered, som TwinSet Limited 12 fast numrerad.

Promo

Promotion (copy), en icke officiellt utgiven utgåva i reklamsyfte för en kommande skiva. Det exakta formatet på denna kan skifta.

7

Sjuan står för antalet tum som denna singels diameter och är mer vad man kallar en "vanlig" singel och innehåller till skillnad från maxisingeln oftast bara två låtar.

aEP

Analog extended play. EP är som ett album fast mindre antal låtar, en till omfånget mindre LP kan man säga. Bokstaven a som står innan är en egen benämning för att här i databasen särskilja mellan analog och digital EP.

mp3

MPEG-1 Audio Layer 3, det nu populäraste ljudformatet för att köpa filer från nätet och lagra på datorer. Det har trots sin komprimering av låtmaterialet en väldigt bra kvalitet och en liten och behändig filstorlek. Detta är det enda formatet som jag tagit med här där ljudfilerna inte är lagrad på ett fysiskt media. Det går sedan självklart att lagra mp3-filer på tex en dvd eller cd-skiva om man så vill.