



Laboration: Flerskiktade applikationer

Målsättning

Syftet med denna laboration är att ge den studerande kunskap och erfarenhet i att utveckla flerskiktade applikationer baserade på ASP.NET MVC och OOP samt att utveckla och konsumera webbtjänster. Ytterligare ett mål är att ge den studerande ett tillfälle att tolka och följa en teknisk kravspecifikation.

Teori

För kursen angiven kurslitteratur.

Material

Visual Studio inklusive SQL Server.

Genomförande

Uppgiften och dess specifikation är omfattande så ta er tid att läsa igenom och försök förstå uppgiften innan ni sparkar igång med själva utvecklandet. En tillfredsställande lösning på samtliga moment är ett måste för att bli godkänd på denna laboration. De uppgifter som ingår kan lösas individuellt eller i grupper om två studenter. Vissa moment i laborationen kräver samarbete med andra laborationsgrupper.

I denna uppgift erhåller ni en teknisk kravspecifikation som medvetet är vagt formulerad, det finns två bakomliggande tankar till detta. Den första är att det är vanligare än man tror att en beställare av ett system saknar erforderlig beställarkompetens (det är då vanligt att man som utvecklare får kliva in och stötta beställaren genom att förtydliga specifikationen), den andra tanken är att inte vingklippa er utan att ge er ett tillfälle att själva reflektera över vilken lösning som blir den bästa.

För att erhålla plus på denna laboration ska två av fyra fördjupningsuppgifter vara lösta. Dokumentera ert arbete väl, rättande lärare ska inte behöva gissa om rapporten är på plusnivå eller inte!

Redovisning

Redovisning av laboration och eventuella fördjupningsuppgifter sker individuellt. Laboration och eventuella fördjupningsuppgifter redovisas dels genom en individuell rapport som beskriver hur du löst uppgiften/uppgifterna samt genom uppvisande av väl fungerande applikation. Redovisning av fungerande applikation sker i slutet av kursen och tid för denna bokas via e-post/moodle alternativt så spelar ni in en filmsnutt där ni visar att och hur er applikation fungerar.

Uppgift

Uppgiften går ut på att skapa **två** system som ska arbeta tillsammans, systemen som efter komplettering skulle kunna gå i drift på en tidning. Det ena systemet ska hantera annonsering och annonsörer och det andra ska hantera prenumeranter.

Teknisk kravspecifikation

Vi vill att ni utvecklar ett prenumerantsystem som hanterar och lagrar information om tidningens prenumeranter.

Vi vill också att ni utvecklar ett separat annonssystem där annonsörer själva kan lägga in sina annonser. Det finns två typer av annonsörer

1. Prenumeranter som ska få en fördel av att vara prenumeranter
2. Företagsannonser

För att slippa dubbellagrad information ska annonssystemet hämta uppgifter från prenumerantsystemet. Om annonsören finns med i prenumerationssystemet så ska personuppgifter hämtas därifrån.

Kommunikationen mellan annonssystemet och prenumerantssystemet sker via webbtjänst.

Funktionen hos annonssystemet

Detta system ska ha ett webbgränssnitt där man kan lämna in annonser. Det är en sida som gör skillnad mellan prenumeranter och företag. En prenumerant ska bara behöva ange sitt prenumerationssystemet för att kunna lägga till en annons. Systemet hämtar då uppgifter som namn, telefonnummer, utdelningsadress, postnummer samt ort från prenumerationssystemet.

Rent praktiskt ska det gå till så här:

1. Man kryssar i en radioknapp för att markera att man är prenumerant.
2. Man skriver i sitt prenumerationssystemet.
3. Systemet konsumerar prenumerationssystemets webbtjänst för att de uppgifter som behövs; Namn, Telefonnummer, Utdelningsadress, Postnummer samt Ort.
4. Hämtade uppgifter visas på webbsidan då de är hämtade.
5. Om någon uppgift är felaktig ska man kunna ändra denna.
6. Efter detta kan annonsen läggas in. Den innehåller då fem fält: id, varans pris, innehåll, rubrik och annonspris.
7. Annonspriset bestäms per automatik till 0 kr för prenumeranter och 40 kr för företag.

Om en företagare vill annonsera används inte en webbtjänst för att hämta information om företaget. Företagaren får fylla i alla uppgifter som behövs direkt i annonssystemet. Då företagaren vill annonsera på webbsidan är det denna procedur:

1. Man kryssar i en radioknapp för att markera att det är en företagsannons.
2. Det dyker upp ett formulär där uppgifter om företaget ska matas in.
3. Det är följande uppgifter: namn, organisationsnummer, telefonnummer, utdelningsadress, postnummer samt ort **plus** fakturaadress som innehåller utdelningsadress, postnummer samt ort.
4. Efter detta kan annonsen läggas in. Den innehåller då fem fält: id, varans pris, innehåll, rubrik och annonspris.
5. Annonspriset bestäms per automatik till 0 kr för en prenumerant och 40 kr för ett företag

För att visa annonserna gör du en sida som visar en lista med annonser. Det ska tydligt framgå om det är ett företag som annonserar.

Uppbyggnad och dokumentation av annonssystemet.

System ska vara flerskiktat med en SQL Server-databas som datalager. Logiklagren bör bestå av ett dataaccesslager samt specialiserade klasser för annonser och annonsörer.

Systemet ska konsumera en webbtjänst som finns i prenumerantsystemet. Detta system bidrar med två saker. Först och främst vill man veta om det är en abonnent som vill annonsera men man vill även få uppgifter, som t.ex. adress om abonnenten. De som inte är abonnenter eller företag får inte annonsera.

All data som skickas mellan de olika lagren i systemet ska hanteras som objekt/modeller. Det kan vara "rena" Entity Framework-objekt (EF-objekt), vy-modeller som utnyttjar EF eller modeller som har helt egenskriven kod som DAL, data-access-lager. Alla modeller ska dokumenteras med UML (eller annan typ av klassdiagram) samt en beskrivning som innehåller en funktionsbeskrivning, in- och ut-parametrar samt returvärdet.

Vi är lite petiga då det gäller databasen. Databasen i detta system ska bestå av två tabeller, tbl_annonser och tbl_ads. tbl_ads ska innehålla annonserna. Vi använder engelska här för att det inte ska uppstå förväxlingar i namnet. De har en relation då en annonsör kan ha flera annonser. Vi vill att ni baserar er på en namngivningspolicy där samtliga tabellnamn börjar med tbl_. Tabellen med annonserna ska heta tbl_ads. De fält som ingår i denna tabell ska börja med ad_, som de två första bokstäverna i advertise. Det betyder att fältet rubrik ska heta ad_rubrik, fältet för innehåll, ad_innehall e.t.c. Tecknen för å, ä och ö får inte förekomma. Alla namn ska skilja sig från varandra så att det aldrig uppstår risk för förväxlingar. Dokumentera databasen med ett ER-schema.

Om man genererar klasserna direkt med EF så blir det ganska dåliga namn på klasser, metoder och egenskaper. Kan man få det snyggt och läsligt på Svenska?

Uppbyggnad och dokumentation av prenumerantsystemet

Detta system ska som namnet antyder hantera tidningens prenumeranter. Det ska bestå av en SQL Server-databas med en tabell innehållande prenumerantens prenumerationsnummer, personnummer, förnamn, efternamn, utdelningsadress och postnummer.

För att spara värdefull tid hoppar vi över själva användargränssnittet i systemets grundutförande. Men systemet ska ha ett gränssnitt i form av en webbtjänst som svarar upp mot de behov som ställs i annonssystemet.

På samma sätt som för annonssystemet ska ni dela upp systemet i lager. Dokumentera på samma sätt som det andra systemet.

OBSERVERA!

Vi pratar om två system med var sin databas. I Visual Studio skapar ni ett projekt för vardera system.

Ni har säkert redan insett att det finns dubbel information i systemen. Abonnentens namn, adress etc. finns både i annonssystemet och i abonnentsystemet. Byter abonnenten adress vet vi inte riktigt hur detta hanteras på bästa sätt. Här vill vi att ni utreder och implementerar det på ett sätt som ni tycker blir bäst.

Fördjupningsuppgifter

Som tidigare nämnts krävs det att ni löser två av följande fyra uppgifter för att erhålla plus på denna laboration.

Säker dataöverföring mellan de två systemen

Utred hur man gör och implementera säker dataöverföring mellan de två systemen.

Utveckla ett enklare gränssnitt mot prenumerantsystemet

Här utvecklar ni en enklare Windowsapplikation som hanterar att lägga till, redigera och ta bort prenumeranter i systemet.

Konsumera någon annan students webbtjänst

Modifiera ditt annonssystem så att det konsumerar en webbtjänst som någon annan student har utvecklat. Vid redovisning av laborationen ska det finnas två versioner, en där du konsumerar din egen webbtjänst och en där du konsumerar någon annans.

Import och exportfunktion till det ena eller andra systemet

Utveckla en funktion som genererar och sparar all information runt annonser eller prenumeranter i en XML-fil. Utveckla även en importfunktion där systemet kan laddas med prenumeranter eller annonser genom inläsning från en XML-fil.