Linnéuniversitetet
Kalmar Växjö

Laborationsanvisning

Individuellt arbete

Steg 2, laborationsuppgift 1

Linnéuniversitetet
Kalmar Växjö

Författare: Mats Loock
Kurs: ASP.NET MVC
*Kurskod:*1DV409

Linnéuniversitetet
Kalmar Växjö

Upphovsrätt för detta verk

Detta verk är framtaget i anslutning till kursen ASP.NET MVC (1DV409) vid Linnéuniversitetet.

Du får använda detta verk så här:

Allt innehåll i detta verk av Mats Loock, förutom Linnéuniversitetets logotyp, symbol och kopparstick, är licensierad under:

Creative Commons Erkännande-IckeKommersiell-DelaLika 2.5 Sverige licens.
http://creativecommons.org/licenses/by-nc-sa/2.5/se/

Det betyder att du i icke-kommersiella syften får:

- kopiera hela eller delar av innehållet
- sprida hela eller delar av innehållet
- visa hela eller delar av innehållet offentligt och digitalt
- konvertera innehållet till annat format
- du får även göra om innehållet

Om du förändrar innehållet så ta inte med Linnéuniversitetets logotyp, symbol och/eller kopparstick i din nya version!

Linnéuniversitetet
Kalmar Växjö

Innehåll

Linnéuniversitetet
Kalmar Växjö

Inledning

Kursens avslutande moment utgörs av ett individuellt arbete som ska resultera i en databasdriven webbapplikation. Det individuella arbetet ska utföras individuellt, men du uppmuntras att diskuterar eventuella problem och lösningar med andra kursdeltagare.

Det är viktigt att du följer anvisningarna till det individuella arbetet.

Viktiga datum och klockslag

- **2013-12-10 13:15**

  Förutsättningarna för det individuella arbetet presenteras.

- **2013-12-17 12:00**

  Om du inte väljer att göra en väderapplikation är detta den senaste tidpunkten för inlämning av kort beskrivning av det individuella arbetet.

- **2014-01-14 12:00**

  Senaste tidpunkten för publicering på kursens webbhotell av den webbapplikation du muntligen redovisar under torsdagen den 17 januari eller fredagen den 18 januari 2013.

- **2014-01-16 - 2013-01-17**

  Muntlig redovisning av det individuella arbetet.

Resurser

För att skydda de resurser du behöver tillgång till under det individuella arbetet är dessa placerade i ett virtuellt privat nätverk (VPN). Du måste skapa en VPN-anslutning till vpn200.lnu.se för att komma åt dessa resurser (se bilaga 1).

Resurserna består av en webbserver och en databashanterare. Dessa utgör det webbhotell där du publicerar din webbapplikation. Webbhotellet tillhandahåller en applikationsrot (placerad i en egen applikationspool) samt en databashanterare där du själv får skapa den databas/de databaser du behöver.

På kursens webbplats hittar du de autentiseringsuppgifter du behöver för att kunna använda webbhotellet.

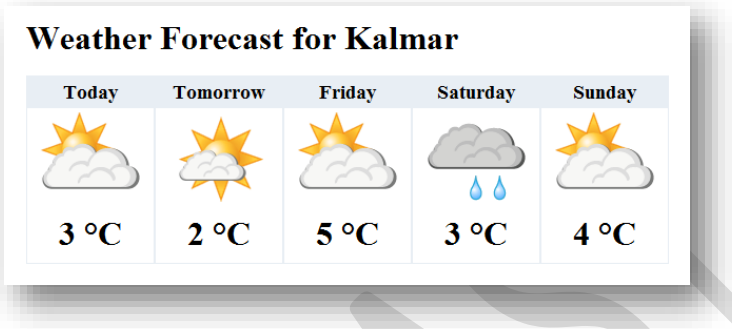
UTKAST

Linnéuniversitetet
Kalmar Växjö

Rekommenderad applikation

Du rekommenderas att skapa en väderapplikation, men du kan välja att skapa en helt annan applikation så länge som alla krav uppfylls. Väljer du att inte skapa en väderapplikation måste du lämna in en kort beskrivning av applikationen.

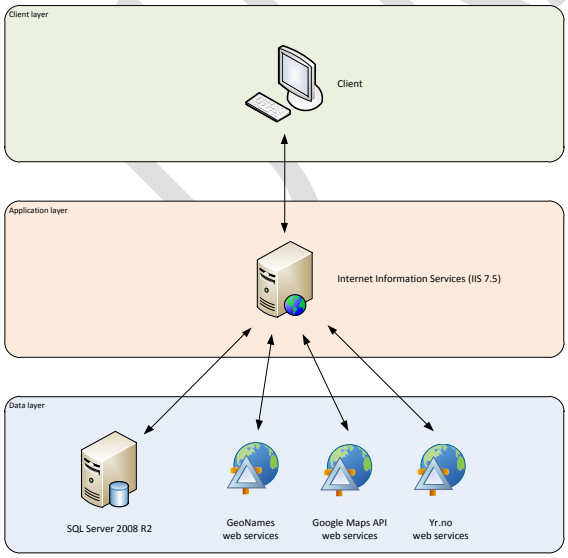
I väderapplikationen ska en användare kunna få en femdygnsprognos presenterad för den plats användaren anger. Finns det flera platser med samma namn ska användaren från en lista kunna välja den plats som prognosen ska visas för.



Figur 1. Exempel på femdygnsprognos för en plats.

Genom att använda minst en webbservice för geografisk information ska webbapplikationen kunna slå upp den nödvändiga information som krävs för att erhålla en prognos från yr.no. Såväl geografisk information som prognoser från olika webbservice ska lagras i en databas för att undvika upprepade, onödiga och tidsödande förfrågningar till olika webbservice.

Figur 2 visar översiktligt väderapplikationens arkitektur där du bland annat finner information som leder till lämpliga API:er att använda.



Figur 2. Väderapplikationens arkitektur.

Linnéuniversitetet
Kalmar Växjö

Krav

Du är fri att skapa och utforma vilken webbapplikation du vill. Du ansvarar för att se till att din applikation uppfyller de krav kursledningen ställer på den beträffande grundläggande tekniker och funktionalitet. Avvikelser från kraven måste dokumenteras och godkännas av kursledningen innan du kan bortse från ett eller flera av dem.

OBS! Väljer du att göra det individuella arbetet som en del av ett examinerade moment i annan kurs ansvarar du för att se till att den kursens krav följs. Uppstår konflikt mellan de olika kursernas krav är det ditt ansvar att ta upp och diskutera det med ledningen för respektive kurs.

Betyg 3

Kommenterad [ML1]: Komplettera med krav på att modellen ska placeras i ett separat projekt, en domänmodell.

För betyget 3 på det individuella arbetet ska din applikation uppfylla nedanstående.

1. Applikationen ska arbeta med data från minst två datakällor, vara av en måste vara en databas. En databas med minst en tabell och hämtning av data från minst en webbservice anses uppfylla kravet på två datakällor.
2. Speciella krav beroende på typ av applikation.
   a) Väljer du att göra en väderapplikation måste följande krav vara uppfyllda:
      1. En femdygnsprognos ska presenteras. En dags prognos ska minst baseras på period 2, d.v.s. prognosen mellan 12:00-18:00. Saknas period 2 ska period 3 användas. Saknas period 3 tas nästa dags period 2. Det står dig fritt att utöka presentationen av prognoserna och presentera prognoser för samtliga perioder eller prognoser för varje timme.
      2. En prognos ska minst innehålla en bild beskrivande vädret samt temperatur. Det står dig fritt att utöka presentationen av en prognos med ytterligare information som t.ex. vindriktning, vindhastighet och nedbördsmängd.

   b) Väljer du inte att göra en väderapplikation måste du lämna in en kortfattad beskrivning av applikationen om maximalt en A4-sida innehållande syftet med applikationen samt en konceptuell modell över databasen.
3. Webbapplikationen ska vara skapad med Microsoft ASP.NET MVC 5 och C#.
4. Relationsdatabasen SQL Server 2008 R2 ska användas för persistent lagring av data.
   a) All kommunikation med databashanteraren ska ske genom användaren **appUser** med lösenordet **1Br@Lösen=rd?**.
5. Webbapplikationen ska följa riktlinjer för utformning av innehåll på webben enligt W3C WCAG 2.0.
6. Webbapplikationen ska vara uppdelad enligt designmönstret Model –View –Controller.
7. För hantering av persistent data ska Microsoft Entity Framework användas.
8. Controllermetoder får inte använda sig direkt av några objekt härrörande från Entity Framework, utan måste använda sig av ett centrallager (”repository”) och servicelager.
9. Webbapplikationen ska ha CRUD-funktionalitet (”Create”, ”Read”, ”Update”, ”Delete”), d.v.s. användaren ska, förutom att kunna skapa nya poster, även kunna läsa, redigera och ta bort befintliga poster i tabeller i en databas. Avsteg från detta krav kan i undantagsfall godtas beroende på implementation.
10. Användaren appUser ska:
    a) ha rättigheter att exekvera SQL-satser för SELECT, INSERT, UPDATE och DELETE.
    b) ha rättigheter att exekvera lagrade procedurer (Execute).

c) inte ha några andra rättigheter till andra databasobjekt.

11. Applikationen ska:
    a) ha en hög användbarhet, d.v.s. vara lätt att lära sig, effektiv att använda och ge positiva upplevelser för användaren.
    b) ha en bra och genomarbetad layout och design.
    c) tillhandahålla en genomtänkt och logisk navigation.
    d) använda sig av minst en " Layout Page".
12. Allt data ska valideras. (Validering i "controller layer" inte tillåten!)
13. Validering av formulärdatat ska vara utformad så att valideringen sker i så stor utsträckning som möjligt på klienten innan datat skickas till servern för validering och berarbetning.

Betyg 4

För betyget 4 på det individuella arbetet ska din applikation, förutom kraven som ställs för betyget 3, uppfylla nedanstående:

1. Leveransdatum ("deadlines") ska hållas; det är en förutsättning för ett högre betyg än godkänd (3).
2. Minst ett textfält i applikationen ska ha ”autocomplete” och controllermetoden som klienten anropar ska returnera data av typen `JsonResult`.
    a) Väljer du att göra en väderapplikation måste följande krav vara uppfyllda:
        1. Underlaget för de platser som presenteras av textfältet med ”autocomplete” ska hämtas från databasen och får inte hämtas från en extern webservice.
3. Om det finns flera platser med samma namn ska en karta presenteras där de olika platserna markerats.
4. En karta där aktuell plats markerats ska presenteras i anslutning till en prognos.
5. Controller- och/eller serviceklasser måste använda sig av DI (”*dependency injection*”) för att möjliggöra tester.
6. "Test-Driven Development" (TDD) ska ha undersökts. Det ska finnas minst fyra testmetoder i ett separat testprojekt som testar funktionaliteten i fyra olika controllermetoder.

Betyg 5

För betyget 5 på det individuella arbetet ska din applikation, förutom kraven som ställs för betyget 4, uppfylla nedanstående:

1. Leveransdatum ("deadlines") ska hållas; det är en förutsättning för ett högre betyg än godkänd (3).
2. Code First ska tillämpas, d.v.s. klasserna `DbContext` respektive `DbSet<T>` ska användas utan att du använder ”ADO.NET Entity Data Model” (.edmx-fil). OBS! Detta krav medför att appUser måste ha lämpliga rättigheter för att kunna exekvera SQL-satser för INSERT, UPDATE och DELETE.
3. För hantering av webbapplikationens användare och roller ska fördefinierade tabeller och lagrade procedurer kopplade till Microsofts API:er för "Simple Membership" och "Roles Manager" användas.

    Exempel på funktionalitet applikationen ska erbjuda listas nedan. Visar det sig att listan inte är tillämpbar på din applikation diskutera det i så fall med kursledningen.

    a) Användare ska kunna autentiseras.
    b) Användares åtkomst av resurser ska styras med hjälp av roller.

UTKAST

c) En användares autentiseringsuppgifter ska kunna skapas, antingen genom självregistrering eller genom att en administratör gör det.

d) Användare ska kunna logga in och logga ut.

e) Användare ska kunna ändra sitt lösenord.

f) En lista med de fem senaste platserna användaren begärt en prognos för ska kunna visas för användaren. Användaren ska kunna ta bort en plats ur listan.

UTKAST