



INLÄMNINGSUPPGIFTEN

Gruppuppgiften + Enskild uppgift



WEBSHOPPEN 2

Gruppuppgiften

Webbshoppen

- Ni ska bygga en webbshop som säljer ett valfritt sortiment.
 - Webbshoppen ska ha följande funktionalitet
 - Startside
 - Välkomst-text
 - Tre utvalda produkter ska visas på sidan. Detta ska kunna anges i admin.
 - Shopsida med minst tre kategorier
 - T ex Tröjor, Byxor och skor(om ni gör en klädes-shop), cirka 5 produkter av varje.
 - Möjlighet att fritextsöka
 - Varje produkt ska kunna väljas, för att få veta mer om produkten
 - Varje produkt ska ha ett val för "köp" som adderar produkten till kundkorg
 - Övrig info: Kort text om produkten, Pris

Webbshoppen

- Varukorg
 - Valda produkter visas i lista
 - Möjligt att ändra antal
 - Möjlighet att ta bort product
 - Priset visas och summan av produkterna visas längst ner
 - Frakt-vy
 - Namn, adress m m ska gå att fylla i via `console.readline`.
 - Val av frakt med minst två alternativ, med olika pris
 - Betala-vy
 - Produkterna visas, med pris
 - Pris med frakt, samt moms visas.
 - Val av betalningsmetod, minst två. Betalningen behöver självklart inte vara på riktigt
 - När varan är betald så töms varukorgen

Webbshoppen - Admin

- Admin
 - Produkter – kunna lägga till nya, ta bort och ändra
 - Produktnamn
 - Infotext
 - Pris
 - Produktkategori
 - Leverantör
 - Lagersaldo
 - Produktkategorier
 - T ex byxor, tröjor osv
 - Kunder
 - Ändra uppgifter om kunden
 - Beställningshistorik

Förslag på datafält

- Produktnamn
- Pris
- Detaljerad information
- Produktkategori
- Produktleverantör
- Utvald produkt (bool)
- Leveransalternativ
- Kund
 - Namn
 - Adress
 - Gata
 - Stad
 - Land
 - Telefon
 - Mejl
 - Ålder
- Data avseende betalkort
- Summa kostnad för kunden
- Betalsätt
- Utförda köp, baserat på kund, kategorier m m
- Fundera på relationer, kardinalitet, constraints, normalisering osv...

Förslag på querys

- Bäst säljande produkter
- Populäraste kategorin
- Populäraste produkt för herr/dam-sortiment
- Populäraste produkt i olika åldersgrupper
- Flest beställningar per stad
- Försäljning sorterat på leverantörer

Teknik

- Fokus ligger på databasdelen, men programmeringen ska vara av god kvalitet, (DRY)
- En viss del av uppgiften ska vara utförd med ren SQL, t ex via Dapper
- En betydande del ska göras med hjälp av Entity Framework, inkl constraints
- LINQ ska användas, men mer komplexa queries kan göras med SQL/Dapper.
- Databasen ska ligga på Azure
- Dokumentdatabas (MongoDB) samt transaktioner behöver **inte** vara med i projektet
- Övrig Teknik
 - **Asynkrona anrop** till databasen, inkl tidtagning
 - Moment som Enums, try/catch ska finnas med.

Redovisning och inlämning

- Redovisning sker 19/1 muntligen, där funktion och struktur beskrivs
- Butiken ska vara fylld med produkter, tidigare kunder och inköp
- Inlämning sker 19/1
 - Inlämningen består av
 - ER-diagram (Lucidcharts)
 - GIT-länk till projektet (Console-appen)
 - Testdata ska finnas inlagt
 - Databasen ska finnas upplagd på Azure
 - Specifikationen kommer eventuellt att ändras/specificeras (Agil utveckling)

Tidigare uppgiften

- I kursen agil utveckling byggde ni också en webbshop, dock bara i form av grafik.
- I den här uppgiften bygger ni istället funktionaliteten, utan grafik(Bara konsol-text)
- Kanske kan ni ha nytta av hur ni tänkte under den uppgiften...

Arbetsgång

- Diskutera
- Skissa
- Rita I t ex Lucidchart
- Skapa klasser
- Programmera funktionalitet
- Testa (Funktionstest)
- Planera redovisning

Enskild uppgift - VG





Bokningsappen

- En app för att kunna boka t ex olika konferensrum, prylar eller ett bord på en restaurang.

Bokningsappen - Grundkrav

- En person som vill boka ska kunna ange:
 - Namn
 - Adress/avdelning på företaget
- En bokningsansvarig ska via ett admin kunna lägga in ett antal olika object.
- Exempel på ett objekt:
 - Konferensrum (med info om rumsnummer/rumsnamn, antal platser, faciliteter: projector, whiteboard...),
 - Skidor och pjäxor för uthyrning (Storlek, märke, typ av skidor)
 - Tvättstuga...
 - Valfritt annat object, som normal bokas, med en handfull egenskaper.

Bokningsappen - Grundkrav

- I consollen ska man kunna se vilka dagar som objektet är ledigt och upptaget, och det ska bara gå att boka de lediga tiderna.
- Exempel:
- Vecka: 52 [Ä. Ändra vecka]

	Rum 1	Rum 2	Rum 3...
Måndag	[1]	[2]	[3]
Tisdag	Johansson	Nilsson	[6]
Onsdag	[7]	[8]	Bergman
Torsdag	[10]	[11]	[12]
Fredag	Ali	[14]	[15]

Bokningsappen – Admin

- Det ska finnas ett admin, där man kan lägga till nya objekt
 - Namn på objektet, eventuella egenskaper, såsom Namn, antal platser, plats, storlek osv...
- Här ska dessutom de queries du valt finnas valbara i en meny.

Queries

- Exempel:
 - Populäraste objektet/rummet
 - Person som bokat flest gånger
 - Procent ledigt/upptaget per vecka + totalt
 - Egna förslag...

Teknik

- Skapa minst två tabeller, med constraints mellan person och bokningsbart objekt.
- Bygg allt enligt Entity framework och Code first
- Databasen ska ligga på Azure
- Queries med hjälp av LINQ

VG-del

- Målet med VG-uppgiften är att: **“Den studerande visar förmågan att lösa programmeringsrelaterade uppgifter på ett genomtänkt sätt som påvisar djupare förståelse för kodens uppbyggnad. Den studerande kan planera sitt arbete så att leverans sker enligt deadline.”**
- Här har du möjlighet att visa framfötterna, och bygga en applikation som visar på din kunskap och förmåga.
 - Extra features + högre krav på användning av databaser och kodning.
 - Du väljer själv vilka features du vill visa upp, och **väljer teknik efter vad som är lämpligt (kommentera i koden hur du tänkt)**.
- Om du väljer att göra VG-delen så ska den också redovisas inför klassen, muntligt fredagen den 19/1
- Viktigt är att bara göra VG-uppgiften om du själv känner att du uppfyller kursens krav på VG.

VG-delen

- Exempel på extra features:
 - Förutom enkel bokning, av enskilda entiteter ska det i VG-delen också gå att boka flera olika saker (t ex både bord och föreställning på teatern), kunna ange klockslag, och speciella egenskaper, såsom bokning av ett speciellt bord, en speciell plats på en teater.
 - Om du väljer t ex en teater ska det också gå att välja en specific föreställning så det går att se saker som att första föreställningen har 97 lediga platser, andra föreställningen har 45 lediga platser.
 - Möjlighet att sätta pris på bokningarna
 - Inloggning, så du kan demonstrera minst två användares upplevelse
 - Möjlighet att boka av, boka om ska finnas
 - Möjlighet att se vad just den användare man är inloggad som, har bokat.
 - Stor möjlighet att själv bestämma vad du tror bäst visar din kompetens...

VG-delen

- Större vikt på komplexitet/constraints/normalisering osv
- Korrekta namn på databaser/klasser/properties/listor osv
- Välj att göra koden självförklarande, men kommentera där det behövs.
- Se till att applikationen är extra enkel och logisk att använda
- Skapa intressanta och givande queries.

Inlämning Bokningsappen

- Inlämning sker via skolans platform, senast den 19/1, som GIT-länk.
- Applikationen ska gå att köra i befintligt skick, med all data lagrat i databas på Azure.
 - Se till att databasen redan har lite data/bokningar
- Se till att läraren har tillgång till att använda databasen (IP-adress: 178.16.211.2)
- VG-uppgifterna ska också redovisas muntligen.

Slutord om uppgifterna

- Två uppgifter
 - Grupp-uppgift Webbshoppen
 - Enskild uppgift Bokningsappen (VG-nivå)
- Webbshoppen är den viktigaste inlämningen, och där avgörs om det är G på kursen.
 - Om gruppen lämnar en extra bra webshop kan det till viss del påverka bedömningen av eventuella VG-inlämningar. Det kan också fungera på motsatt sätt ;-)
- Eftersom webbshoppen är en gruppuppgift är det extremt viktigt att alla gruppmedlemmar bidrar och är med hela tiden. (Må-fr, 9-16)
 - Arbete med enskilda uppgiften görs på tider ni i gruppen själva bestämmer.
 - Om någon behöver vara frånvarande, stäm av med Gruppen och läraren.
- All kod som levereras i webbshoppen ska samtliga gruppmedlemmar förstå och kunna förklara.
 - Jobba främst med mob-programmering, och turas om att vara den som skriver.

