

CASE - Bankapplikation i Azure = publik sajt



Leif Stefan Holmberg • 13 mars (Redigerad 13 apr.)

100 poäng

Inlämning 12 maj

Administration för banksystem med "stor" databas

Allmänt

Detta är en individuell uppgift. Du ska bygga ett administrationssystem för en bank.

Fusk

Uppgiften ska lösas individuellt. Vid redovisningen får du frågor och måste beskriva hur ditt program fungerar så du måste kunna förklara hur all din egen kod fungerar.

Det är inte tillåtet att plagiera (skriva av) någon annans kod. Även om du byter namn på klasser, metoder och variabler är det enkelt att se om två olika applikationer har samma struktur.

Betyg

Det går att få G, VG eller IG på uppgiften. Den är obligatorisk tillsammans med inlämningsuppgift 1 för att kunna få G på hela kursen. För att få G på inlämningsuppgiften ska alla krav märkta med G vara uppfyllda. För att få VG på inlämningsuppgiften ska alla krav märkta med G och VG vara uppfyllda.

Redovisning

Privat github bjud in mig som admin/collaborator (stefan.holmberg@systementor.se) så jag kan se kod i repo(s). Skriv in länk till ert repo i mappen Inlämningsuppgift nedan - det är där jag kopplat repo till person och betygsätter. Jag ska kunna ladda ner **er kod** och koppla projekt mot min (startdatabas enligt nedan) och vid F5 ska applikationen starta, migrera osv och bli körbar . Dessutom sajten körbar i Azure!

Krav för godkänt



Du ska i grunden utgå från den databas (DbContext + seed data) som ingår i startprojektet. .

Bygg ett webbgränssnitt med ASP.NET Core Razor Pages. Ladda ner en "FREE BOOTSTRAP TEMPLATE" och modifiera layout etc så den används! Alt skapa CSS från grunden själv. Men jag vill att sajten ska se "bra" ut!

Grundprinciper kring "best practices", framförallt ViewModels . Jag vill inte se någon databasentitet i vyer

Vissa saker nämner jag inte specifikt, tex förutsätter jag att du implementerar inputvalidering etc. Detta är ett RIKTIGT system med "dumma" användare. Att du programmerar mot Interface och inte faktiska implementationer är en annan självklarhet

Startsida: skriv ut statistik på startsidan. Du ska se antal kunder, antal konton och summan av saldot på konton.

Det ska gå att ta fram en kundbild genom att ange kundnummer. Kundbilden ska visa all information om kunden och alla kundens konton. Kundbilden ska också visa det totala saldot för kunden genom att summera saldot på kundens konton.

Det ska gå att söka efter kund på namn och stad. En lista ska visas med kundnummer och personnummer, namn, adress, city som sökresultat. Sökresultatet ska vara paginerat (50 resultat i taget och så ska man kunna bläddra till nästa/tidigare sida). Klickar man på en kund ska man komma till kundbilden.

Lägg till en kontobild där man kan se transaktioner för ett individuellt konto. När man klickar på ett kontonummer i kundbilden ska man komma till en kontosida som visar kontonummer och saldo samt en lista med transaktioner i descending order. Om det finns fler än 20 transaktioner ska JavaScript/AJAX användas för att ladda in ytterligare 20 transaktioner när man trycker på en knapp längst ned i listan. Trycker man igen laddas 20 till, och så vidare.



Det ska framgå tydligt om någon försöker ta ut eller överföra mer pengar än vad som finns på kontot!

ASP.NET Core Identity. Du ska lägga in roller via seed, Rollen Admin ska kunna administrera användare /att administrera användare är VG krav, men gör en actionmetod och tom vy för detta för G fast med rätt behörighet). Rollen Cashier ska kunna administrera kunder och deras konton. OBS: kunder är inte users det är endast admins och cashiers som jobbar med systemet

Rollerna Admin och Cashier ska automatiskt skapas i databasen vid uppstart om de inte finns. (Seed)

Det ska finnas Unit Test-projekt där du skriver tester som testar att det

- 1) inte går att ta ut eller
- 2) överföra mer pengar än det finns på kontot. Det ska inte heller
- 3) gå att sätta in eller
- 4) ta ut negativa belopp.

Det ska också finnas tester som verifierar att det skapas en korrekt transaktion.

Seeda två användare:

stefan.holmberg@systemmentor.se och Hejsan123# och som Admin

stefan.holmberg@customer.banken.se och Hejsan123# och som Cashier

Driftsätt din lösning och databas till Azure.

Krav för VG

Lösningen måste vara gjord med god arkitektur.

Det ska gå att skapa en ny och ändra en kund. När en kund skapas ska de få ett nytt unikt kundnummer. Det ska också automatiskt skapas ett transaktionskonto med kunden som ägare. OBS: inputvalidering (även i browser)

Unittester på controlleraction(s). Gör tester på action för "Visa kundbild".

Unittester på ett repository alt. en klass som använder DbContext. Du ska använda EF Core Inmemory-databas för de testerna



SÖK: Inte mot databasen...En extern söktjänst (Azure Cognitive Search) används som sökmotor. <https://docs.microsoft.com/sv-se/azure/search/search-get-started-dotnet>. Så du måste göra en batchrutin för att fylla index från databasen initialt. Gör detta som en separat console applikation. Samt såklart kontinuerligt skicka över ändrad data (när man tex byter namn i admindelen).

Sökresultatet ska vara möjligt att sortera. Första klicket på en kolumn = asc, sen desc, sen asc etc etc.

Startsidan ska gruppera antal kunder, antal konton och summan av saldot på konton per LAND. När man klickar på tex "Norway" ska man komma till en sida där de 10 största (högst totalsaldo) kunderna i det landet visas. Denna sida ska responsecachas (Vary på land) i en minut.

Banken har också en mobilapp för kunder. Skapa ett Web API till appen.

Ett anrop till /api/me ska ge samma information som i kundbilden.

Ett antop till /api/accounts/12345 ska visa transaktioner för ett konto. Använd parametrar limit och offset för att begränsa antal transaktioner som hämtas. Dessa värden skickas in i LINQ:s Take och Skip.

Använd en Authorization header för att identifiera kunden. Du kan använda basic authentication eller jwt tokens.

Banken har också en Console app (batch som körs varje natt). Det kollar efter misstänkta transaktioner (penningtvätt tex)

Implementera denna så den återanvänder kod. Programmet ska för varje land

 för varje användare i det landet

 för varje kontotransaktion för användaren kolla om den uppfyller regel för "misstänkt"

(se nedan)

 när "landet" är klart skicka en rapport (mail) till <land>@testbanken.se. Bara en lista över vilka personer och vilka kontonummer och vilka transaktionenummer

Håll reda på var du slutar sas så du inte börjar från början varje dag

REGLER: en enskild transaktion större än 15000 kr, Eller totala transaktioner de senaste tre dygnen (72h) från aktuellt tidpunkt större än 23000

DET KOMMER EN STARTMALL HÄR MED PROJEKT ATT UTGÅ FRÅN MED SEEDNING I

Ditt arbete

Tilldelad



Markera som klar



Privata kommentarer

[Lägg till en kommentar på Leif Stefan Holmberg](#)



Klasskommentarer

[Lägga till en klasskommentar](#)

