Administration för banksystem med ”stor” databas

**Redovisning**

Skapa en privat github och bjud in mig som admin/collaborator (RichardChalk) så jag kan se din kod i repo(s). Det är där jag kopplar repo till person och betygsätter.

Jag ska kunna ladda ner **er kod** och koppla projektet mot min egen databas och vid F5 ska applikationen starta, migrera osv. och bli körbar **utan error**.

Dessutom ska sajten (endast webappen) vara körbar i Azure! Inkludera Azure länken i din repository (tex. ReadMe.md).

Databasen finns att ladda ner här:  
[Ladda ner](https://aspcodeprod.blob.core.windows.net/school-dev/BankAppDatav2%20(1).bak)

Efter nedladdningen kör denna syntax på databasen  
alter authorization on database::"database-name" TO [sa]

**Krav för godkänt**

* **Självständigt** bygg en ASP.NET Core lösning för en bank som kan hantera kunder, deras konton med saldo samt transaktioner.
* Du ska använda en databas som tillhandahålls i denna fil i form av länk med data (database first). Använd Entity Framework Core och skapa entiteter och ett DbContext som passar databasen.
* Bygg ett webbgränssnitt med ASP.NET Core Razor Pages (ni får använda även MVC om ni är trygga i miljön och kan arbeta **självständigt**). Ladda ner en ”FREE BOOTSTRAP TEMPLATE” och modifiera layout osv. så den används! Jag vill att sajten ska se ”bra” ut.
* Följ 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 automapper, klassbibliotek, ReadMe.md och inputvalidering osv. Detta är ett RIKTIGT system med "dumma" användare.
* Startsida: Skriv ut statistik på startsidan. Du ska se antal kunder, antal konton och summan av saldot på konton (denna sida är publik och kräver inte authentication).
* 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 (datum). 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.
* Systemet ska också hantera **insättningar**, **uttag** och **överföringar** mellan konton. Det får inte bli några avrundningsfel så använd decimal. Det ska alltså inte gå att ändra saldo direkt på ett konto - alltid via en transaktion.
* 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. Rollerna Admin och Cashier ska automatiskt skapas i databasen vid uppstart om de inte redan finns.
* Rollen **Admin** ska kunna administrera användare /att administrera användare är VG krav (men gör en 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

Seeda två användare:

[richard.chalk@systementor.se](mailto:richard.chalk@systementor.se)  
pw: Hejsan123#   
role: Admin

[richard.erdos.chalk@gmail.se](mailto:richard.erdos.chalk@gmail.se)  
pw: Hejsan123#  
role: Cashier

Driftsätt din lösning och databas till **Azure**.

**Krav för VG**

* Lösningen måste vara gjord med god arkitektur (bibliotek, services osv.).
* 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 – client side)
* Startsidan ska vara publik!!! (Behöver inte logga in för att kunna se den)
* Lägg till ett gränssnitt för att CRUDa användare och sätta roller.
* 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. För att response caching ska funka måste även denna sida vara publik (ingen authentication).

**Banken har också en mobilapp för kunder så det behövs ett Web API till appen.**

I samma solution skapa ett Web API.   
Ett anrop till **/api/me/{id}** ska ge samma information som i kundbilden.

Ett anrop till **/api/account/12345** ska visa alla 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**). Tex. **/api/account/{id}/{limit}/{offset}**

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

Implementera denna så den återanvänder kod. Programmet ska köras 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 skapa en rapport (.txt) och sparar till hårddisken. Bara en lista över vilka personer och vilka kontonummer och vilka transaktionsnummer är misstänksamma.

Håll reda på var du slutar så du inte börjar från början varje dag. Endast nya misstänksamma transaktioner ska inkluderas i listan.

REGLER:   
1. En enskild transaktion större än 15,000kr,   
2. Totala transaktioner de senaste tre dygnen(72h) från aktuell tidpunkt är större än 23000kr