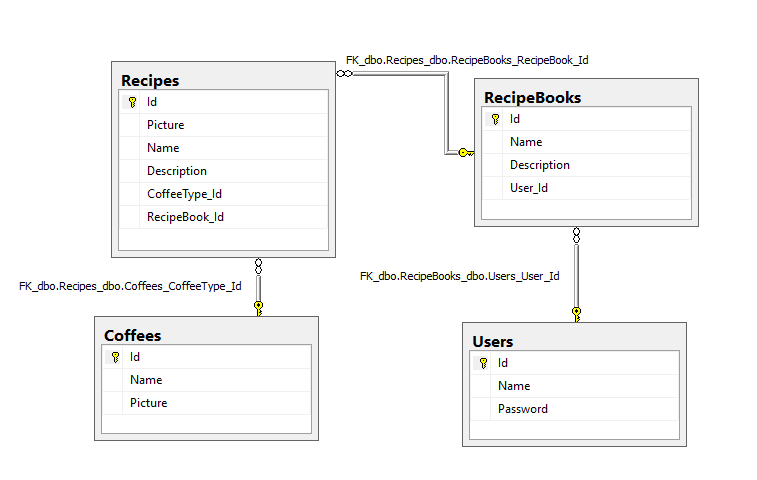
# Rendszerterv

## Adatbázis réteg

**Célja**: a felhasználó által nyilván tartani kívánt adatok tárolása.

Az általunk fejlesztett szoftverben az SQL Server Compact egy külön komponensként, egy NuGet csomag formájában van jelen.



ábra 0.1Az adatbázis sémája grafikusan megjelenítve

## Adathozzáférési réteg

**Célja**: az adatbázisban tárolt adatok leképzése(ORM) C# objektumokra, illetve hozzáférés biztosítása az adatbázishoz az adatok manipulálása érdekében.

Az adathozzáférési réteg alapja a Microsoft Entity Framework, ami lehetővé teszi az előbb említett leképzést a relációs adatbázis rekordjai és C# objektumok (ún. entitások) között.

Az adathozzáférési réteg fő feladatai a következők:

* Adatbázis létrehozása és feltöltése adatokkal
* Hozzáférés biztosítása az adatbázis adataihoz a felsőbb rétegeknek (különösképpen az üzleti logikai rétegnek)
* Az adatbázis tartalmának manipulálása

Az adathozzáférési réteg fő osztálya a **CoffeBookContext** osztály. Ez az Entity Framework Code-First technikájával készült, ami tulajdonképp annyit jelent, hogy először a C# osztályok lettek megírva, amikből utána létrejött maga az adatbázis séma . **CoffeBookContext** osztályon keresztül lehetséges a hozzáférés az adatbázishoz és ez által magukhoz az adatokhoz.

Az adathozzáférési réteg az adatbázis sémájának megfelelően 4 osztályt tartalmaz, melyek az egyes tábláknak feleltethetők meg.

A **Coffee** osztály a kávék adatainak tárolását teszi lehetővé, ami jelen esetben a kávétípus nevét, leírását illetve egy képet tartalmaz.

A **Recipe** entitás az egyes kávékhoz tartozó receptek adatait tartja nyilván. Minden receptnek van egy egyedi neve és kávétípusa, ami a recept alapját képző kávétípus. Tartalmazza még továbbá a recept leírását és egy képet.

A **RecipeBook** osztály a felhasználók összegyűjtött receptjeit tárolja receptkönyvek formájában. A tárolt recepteken kívül rendelkezik még egy névvel és leírással.

A **User** entitás a felhasználók adatait tárolja: a bejelentkezéshez szükséges felhasználónevet és jelszót, továbbá a felhasználó által létrehozott receptkönyveket.

Az adathozzáférési rétegben az adatokhoz való hozzáférés és azok manipulálása tehát a fenti 4 entitás és a CoffeBookContext osztály segítségével lehetséges. Lehetőség van mind a négy entitás esetén:

* Új entitás létrehozására és annak az adatbázishoz adására
* Entitás adatainak lekérdezésére
* Létező entitás adatainak módosítására
* Entitás törlésére és az általa képviselt bejegyzés adatbázisból való eltávolítására

## Üzleti logikai réteg

**Célja**: az adatelérési réteg által szolgáltatott adatokat az adatbázistól független formában továbbítani a felsőbb rétegeknek (főként a grafikus felhasználói felületnek).

Az üzleti logikai réteg az adatelérési rétegre épül. Annak érdekében, hogy az üzleti logikai réteget használó más rétegek ne függjenek az adatátviteli rétegtől az üzleti logikai rétegben az egyes entitások adatainak továbbítására létre lettek hozva úgynevezett DTO (Data Transfer Object) osztályok. Ezek tulajdonképpen elfedik az entitásokat, ezzel lehetővé téve azt, hogy az üzleti logikai réteget használó felsőbb rétegek csak tőle függjenek és az alsóbb rétegektől már ne. Ezek a DTO osztályok a **CoffeeDto**, **RecipeDto**, **RecipeBookDto** és **UserDto**. Az általuk tartalmazott adat megegyezik a megfelelő entitások adattartalmával, viszont már nem az Entity Framework beépített struktúráit használják, ami a megfelelő leképzéshez szükséges, hanem már a .NET keretrendszerbe beépített struktúrákat.

Az üzlezi logikai réteg két fő komponense az **IDbHandler** és az **IDbHandlerFactory** interfészek. Ezek egy egységes felületet nyújtanak az üzleti logikai réteg funkcióinak eléréséhez.

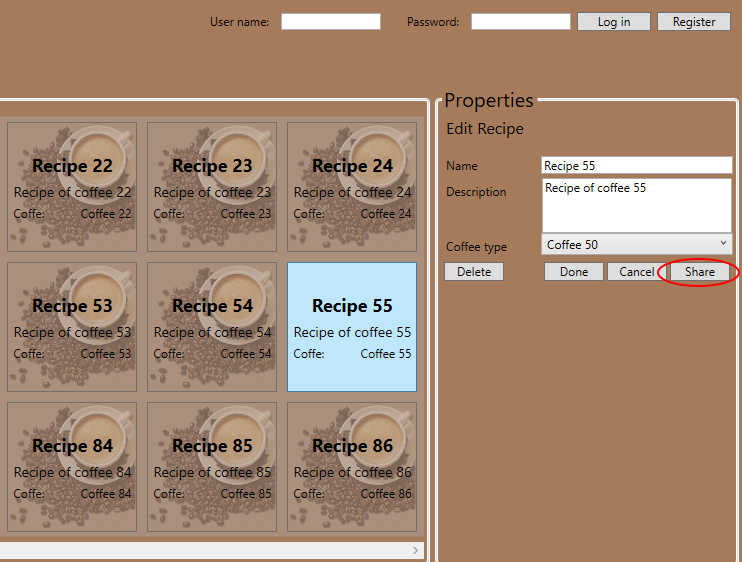
Az **IDbHandler** interfész lehetővé teszi az adatbázis bejegyzéseinek manipulálását, illetve tartalmazza az alkalmazás funkcióihoz szükséges metódusokat, mint például recept hozzáadása receptkönyvhöz. Az ezt az interfészt megvalósító osztály nyújtja üzleti logikai réteg által elérhető funkciókat.

Az **IDbHandlerFactory** mint a neve is mutatja egy factory interfész, egyetlen feladata az **IDbHandler** interfészt megvalósító, épp használt implementációnak a legyártása. Ennek az interfésznek az implementációja van használva a megjelenítési rétegben, ezen keresztül vannak használva az üzleti logikai réteg funkciói.

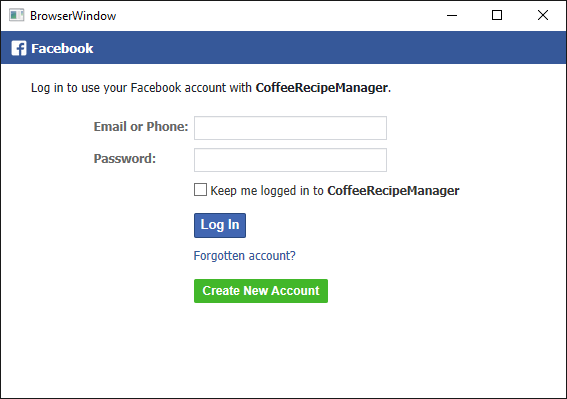
## Megjelenítési réteg

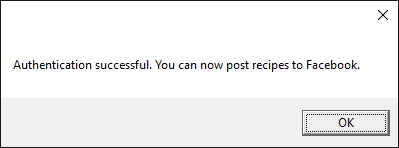
### Receptek megosztása Facebookon

Az alkalmazás lehetőséget ad receptek megosztására a Facebook nevű közösségi oldalon. Ez a funkció az után érhető el, hogy a felhasználó a bal egérgomb egy kattintásával kiválasztott egy receptet. Ekkor megjelenik a kiválasztott recept adatait megjelenítő és módosítását lehetővé tévő oldalsáv a főablak jobb oldalán. A megosztás funkció a **Share** gombra való kattintással érhető el.



A gombra való kattintás után előjön a Facebookos bejelentkezést lehetővé tévő ablak. Itt a felhasználó megadhatja a Facebookos bejelentkezési adatait. A sikeres bejelentkezésről egy dialógusablak tájékoztatja a felhasználót.





A bejelentkezés után az alkalmazás megjegyzi az aktuális Facebook felhasználó adatait, ezek után a felhasználó kijelentkezéséig vagy az alkalmazás bezárásáig ennek a felhasználónak a nevében van lehetőség posztolni. Ennek sikerességéről ismét egy dialógusablak értesít.

