CoffeeBook

Rendszerterv

A Kávé receptkönyv

# Tartalomjegyzék

[A rendszer leírása 3](#_Toc468055032)

[Feladatkiírás 3](#_Toc468055033)

[Biztosított funkciók 3](#_Toc468055034)

[Környezet 3](#_Toc468055035)

[Megvalósítás 4](#_Toc468055036)

[Perzisztencia réteg 4](#_Toc468055037)

[Adatbázis réteg 4](#_Toc468055038)

[Adathozzáférési réteg 5](#_Toc468055039)

[Üzleti intelligencia réteg 6](#_Toc468055040)

[Megjelenítési réteg 7](#_Toc468055041)

[Főképernyő 7](#_Toc468055042)

[Részletező nézet 8](#_Toc468055043)

[Receptek megosztása Facebookon 8](#_Toc468055044)

[Bejelentkezett felhasználó további funkciói 11](#_Toc468055045)

[Megjelenés a kódban 11](#_Toc468055046)

[Rendszerkövetelmények és telepítési leírás 13](#_Toc468055047)

# A rendszer leírása

## Feladatkiírás

A rendszer célja egy olyan többrétegű alkalmazás biztosítása, amely lehetőséget ad a felhasználók számára új kávék, receptek felvételére, azok böngészésére, módosítására, valamint beregisztrált felhasználók számára receptes könyvek készítésére.

A feladat részletes specifikációja a követelményspecifikáció dokumentumban olvasható.

## Biztosított funkciók

Az alkalmazás az alábbi funkciók használatára ad lehetőséget vázlatosan:

* Kávék szerkesztése, melybe beletartozik:
  + Új kávé felvétele,
  + Meglevő módosítása,
  + Meglevő törlése,
  + valamint kávék böngészése, keresése.
* Recept szerkesztése:
  + Új recept felvétele,
  + Meglevő recept módosítása,
  + Meglevő recept törlése,
  + valamint receptek böngészése, keresése.

Minden személynek lehetősége van beregisztrálnia, valamint meglevő regisztráció esetén bejelentkeznie. Bejelentkezés után további funkciók érhetők el:

* Saját receptkönyv szerkesztése:
  + Új receptkönyv felvétele,
  + Meglevő módosítása,
  + Meglevő receptkönyv törlése.

## Környezet

A program futtatása egy vastagkliens alkalmazásként készült el, amely mögött egy adatbázis tárolja el a felhasználó adatait. Az alkalmazásunkat a Windows platformot támogató, WPF technológiára alapuló Single-Page alkalmazásként készítettük el, mely lehetővé teszi a kényelmes és igényes grafikus felhasználói felület kialakítását. Az adatok tárolására az SQL Server Compact-ot használtuk.

# Megvalósítás

A rendszer alapjaiban egy háromrétegű alkalmazásként épül fel, melyet rétegeiben további minták, valamint ismert praktikák segítségével finomítottunk.

A legalsó, vagyis perzisztencia szinten található az adatbázis-, valamint az adatelérési réteg, melyeknek fő feladata az adatok eltárolása, valamint azok közzétételé egy API keresztül a ráépülő alkalmazások számára.

A középső, vagyis üzleti logika szinten azon funkciók vannak, melyek azon túl, hogy lehetőséget biztosítanak az adatbázis elérésére, további segéd függvényekkel támogatják a ráépülő réteg munkáját. Így például az adatok hozzáadása, lekérése, frissítése és törlése (röviden CRUD műveletek), valamint bővített hibakezelés mind ide tartoznak.

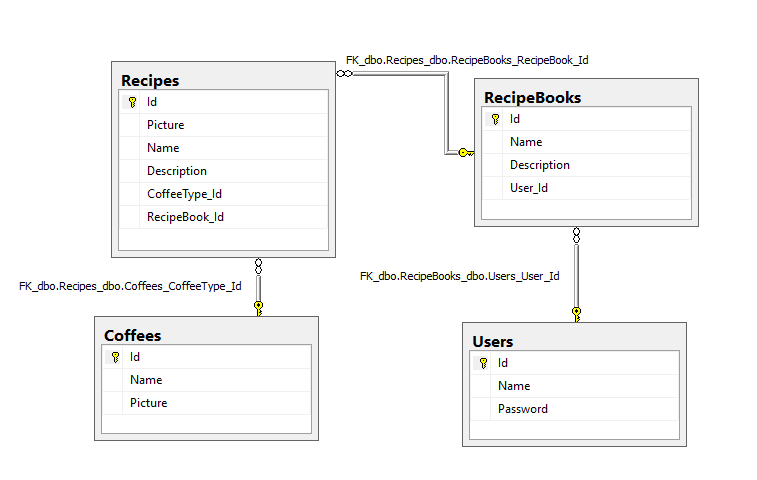
A legfelső, megjelenítési réteg segít a felhasználó és az adatok közötti interakció áthidalására. A WPF technológiára alapuló réteg MVVM (Model-View-ViewModel) mintát követve további három szintre bomlik: View – amely a felhasználói felület kinézetét írja le, Model – amely a felhasználói felületen használatos adatokat, vagyis modelleket írja le, valamint a ViewModel – amely a kinézet és a modell között állva kezeli a felhasználói interakciókat, használja az üzleti intelligencia szolgáltatásait, valamint frissíti mind a modellet, mind a kinézetet a bekövetkezett változtatásoknak megfelelően.

## Perzisztencia réteg

### 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.



### 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 intelligencia 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

**Célja:** Felhasználói felület biztosítása a funkciók használatához.

### Főképernyő

A felhasználói felület egyetlen fő ablakból áll, amely lehetőséget nyújt minden akció elvégzéséhez. A kezdőképernyő három területre osztható: Legfelül a címsáv, ahol lehetőség van beregisztrálni, vagy bejelentkezni; középen a (példa) receptek listája; míg alul a példa kávé típusok listája, ABC szerinti növekvő sorrendben felsorolva.

A címsávban két mező található. Az első a felhasználó nevét, míg a második a felhasználó jelszavát kéri el. A ’Log in’ gombra kattintva lehetőség van bejelentkezni, új felhasználó esetén pedig a ’Register’ gomb ad módot regisztrációra. Hibás művelet esetén a gombok alatt piros szövegként fog megjelenni a hibaüzenet, mely segít a felhasználónak az esetleges probléma (nem található felhasználó, elért mező, stb) kezelésére.

A recept sávban minden adatbázisban található recept megtalálható. A nagy számú receptek miatt lehetőség van azok szűrésére nevük szerint, amelyre a sáv tetején található beviteli mező ad módot. Valamilyen kifejezés keresésekor elegendő csak a kifejezés begépelésére, és a találatok már begépelés közben szűrődnek. Új recept felvételére az ’Add Recipe’ feliratú gomb ad lehetőséget. Egy recept megnyitására elegendő arra rákattintani, amely előhoz egy részletező ablakot (lásd később).

A kávétípusok sávja hasonlóan ad lehetőséget kávétípusok böngészésére, mint receptek esetén.

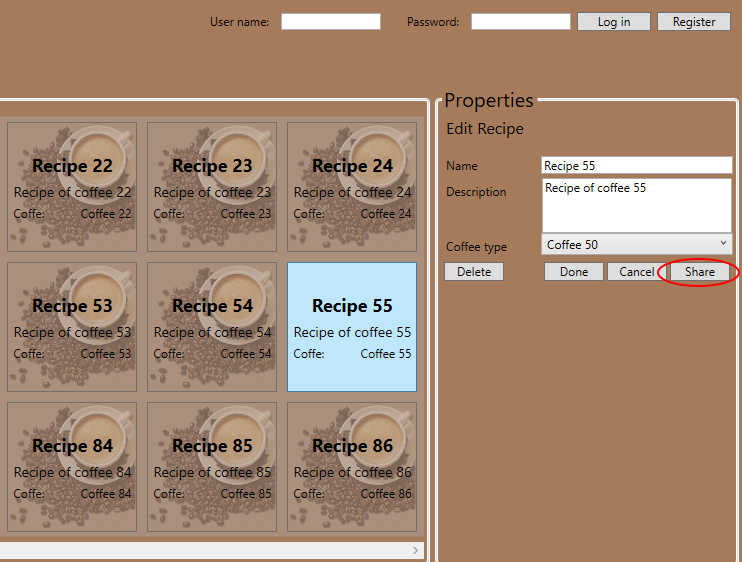


### Részletező nézet

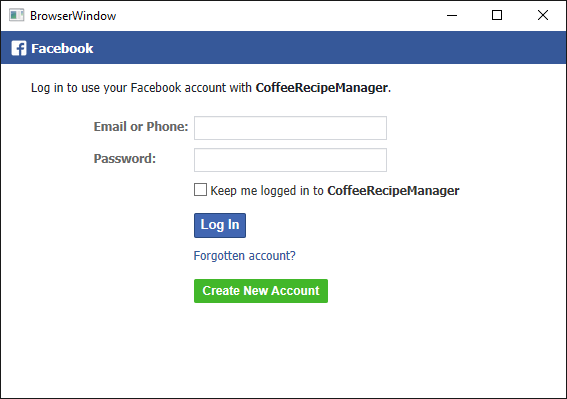
Valamely recept vagy kávé kiválasztásakor vagy új felvételekor a jobb oldalon feltűnik egy Properties felület, amely minden részletét megjeleníti, és szerkeszthetővé teszi. Receptek esetén lehetőség van a recept nevének, leírásának, valamint a kávé típusának megadására és módosítására, míg kávék esetén a kávétípus nevének megadására. Módosítások mentésére a ’Done’ feliratú gombra kell kattintani, Módosítások elvetésére a ’Close’ feliratú gomb ad módot. Továbbá lehetőség van a kiválasztott elem törlésére a ’Delete’ gombbal, valamint receptek esetén mód van annak megosztására is a ’Share’ gombbal.

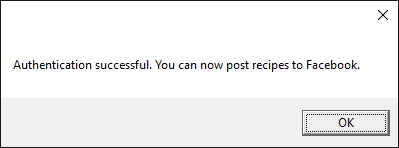
### Receptek megosztása Facebookon

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.

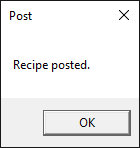


A gombra való kattintás után előjön a Facebookos bejelentkezést lehetővé tévő ablak. Itt a felhasználó megadhatja a Facebookos bejelentkezési adatait. A sikeres bejelentkezésről egy dialógusablak tájékoztatja a felhasználót.





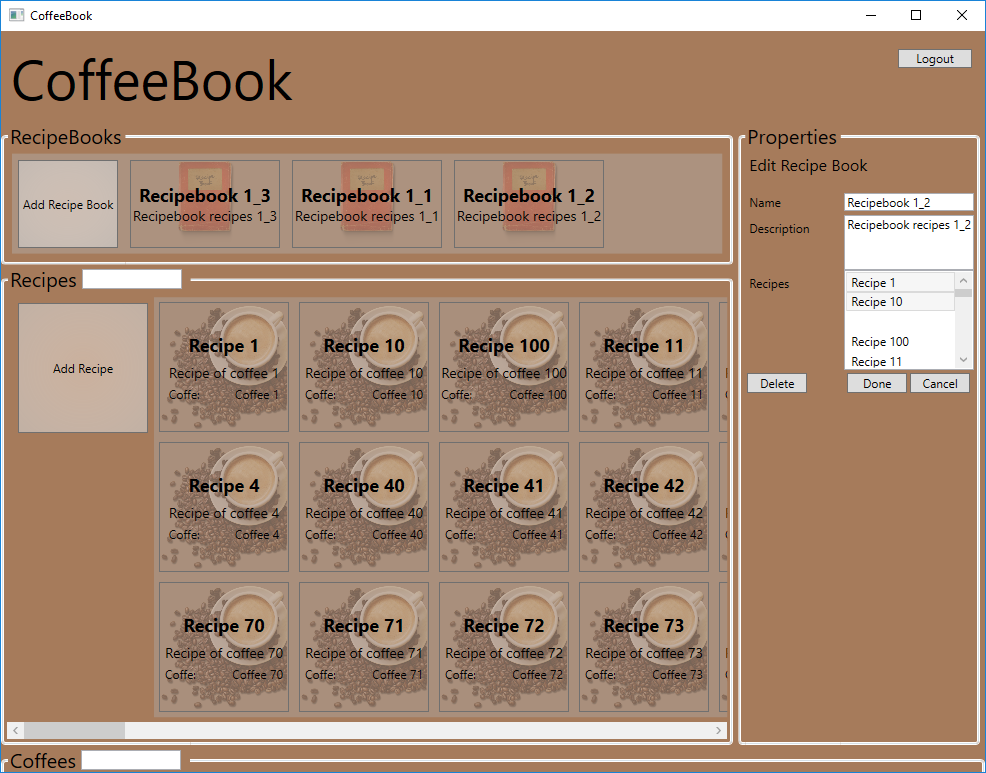
A bejelentkezés után az alkalmazás megjegyzi az aktuális Facebook felhasználó adatait, ezek után a felhasználó kijelentkezéséig vagy az alkalmazás bezárásáig ennek a felhasználónak a nevében van lehetőség posztolni. Ennek sikerességéről ismét egy dialógusablak értesít.





### Bejelentkezett felhasználó további funkciói

Regisztráció, vagy bejelentkezés után egy újabb sáv jelenik meg a receptek fölött, amely a recepteskönyvek számára van fenntartva. Ezeket csak az adott felhasználó látja, és csak saját magának tud újat felvenni, meglevőt módosítani. Új hozzáadásakor vagy meglevő részletezésekor ismét csak a ’Properties’ fül ugrik elő, amely lehetőséget ad a receptes könyv nevének, leírásának, bele tartozó receptjeinek kiválasztására, módosítására. A recepteskönyvhöz tartozó receptek egy üres mezővel vannak elválasztva a nem odatartozóktól.



### Megjelenés a kódban

A felhasználói felület a **CoffeBook** projekten belül található, amely az átláthatóság kedvéért több alkönyvtárra és namespace-re lett bontva. A View könyvtárban találhatóak meg azon xaml fájlok, melyek a felület leírásáért felülősek. A ViewModel könyvtárban található többek között a **MainViewModel**, mely az alkalmazás funkcióit kezeli le, és frissíti mind a modellt, mind a képernyőt. A Model könyvtárban találhatóak azon osztályok, melyek az adatbázisban található adatok felhasználó felülethez igazított változatai. A projekten belül további segéd könyvtárak is megtalálhatóak, melyek a következőek:

* Commands – események kezelésére szolgáló osztályokat tárol.
* Converters – a felhasználói felülethez szükséges átalakító osztályokat tartalmazza.
* Facebook – a Facebook megosztáshoz szükséges osztályokat tartalmazza.
* Helpers – olyan osztályokat tárol, mely az üzleti intelligencia saját modellje, és az MVVM modell eleme közötti átmenetet segíti elő.
* Resources – itt találhatóak olyan statikus állományok, mint például az alkalmazáshoz szükséges képek.

# Rendszerkövetelmények és telepítési leírás

A program elvárható működéséhez minimum az alábbi követelményeket kell teljesítenie az adott rendszernek:

* 800 MHz processzor
* 512 MB RAM
* DirectX 9
* 10 MB szabad tárhely
* .NET Framework 4.6 telepítve

A telepítést a mellékelt csomag setup.exe futtatható állomány elindításával lehet megkezdeni. A feldobott ablakon végigmenve az alkalmazás elérhető lesz a Windows Start menüjén keresztül is. Telepítés után az alkalmazás automatikusan elindul a tesztelést és kipróbálást segítő minta adatbázissal.